

MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA

Cr\$ 10.250

18



Plotter Penman • Digitalização da música
Processador de textos WordStar

rioGráfica

MICROCOMPUTADOR CURSO PRÁTICO

	APLICAÇÕES	
	Nova música eletrônica	357
	SOFTWARE	
	A chave dos dados	360
	CHIPS & BYTES	
	Entradas e saídas	362
	RAIO X	
	Máquina de desenho	365
	SOFTWARE	
	Textos em tela	368
	CIÊNCIA DA COMPUTAÇÃO	
	Circuitos flip-flop	370
	LINGUAGENS	
	Goto end	372
	PERFIL	
	Sord	375

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

- 1) Como são chamados os aparelhos tipo robô, dirigidos por comandos semelhantes aos da linguagem LOGO?
- 2) Qual o nome do dispositivo que coordena diversas linhas melódicas e determina sua duração, criando assim música sintetizada?
- 3) É possível obter, com o processador de textos WordStar, os acentos e cedilhas utilizados em português, sem que apareçam na tela os caracteres de controle?
- 4) O que significa SGBD?

RESPOSTAS DO TESTE ANTERIOR

- 1) A interface que permite a um micro controlar um sintetizador musical é a MIDI (Musical Instrument Digital Interface).
- 2) A variável A! é de precisão simples, enquanto A# é de precisão dupla.
- 3) O software básico que efetua a comunicação entre o sistema operacional em disco e os componentes físicos do computador é o BIOS (Basic Input/Output System).
- 4) Sistema "transparente" é aquele cujas atividades passam despercebidas pelo usuário.

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 30 fascículos de periodicidade semanal. Os 29 primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 30 incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores. As duas primeiras capas são descartáveis; as 3.^{as} e 4.^{as} trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 30 (**Volume 2 e Jogos**).

Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.

Volume 2 - Guardas; frontispício (que virá com o fascículo 30); miolos dos fascículos de 16 a 30; e guardas.

Jogos - Guardas; frontispício e sumário (que constituem as 3.^{as} e 4.^{as} capas do fascículo 30); 3.^{as} e 4.^{as} capas dos fascículos de 1 a 29; e guardas.

rioGráfica

CONSELHO DE ADMINISTRAÇÃO: Roberto Irineu Marinho; João Roberto Marinho; José Roberto Marinho; Oscar Neves; **Diretoria:** Oscar Neves; Danilo Esteves Costa; João J. Noro; Marcos March; Nilo Sérgio de Almeida; **GRUPO EDUCAÇÃO E CULTURA - Divisão Comercial - Diretor:** Jaime Rodrigues; Gerente de Planejamento: João Azeredo Coutinho; **Gerente de Produto:** José Mauro Porto; **Gerente do P.C.P.:** Aylton Menezes; **Diretor de Vendas:** Marcos March; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgar Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto; **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Maurício Rittner; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Editor-assistente:** Claudio Marcondes; **Revisores:** Isis Augusta Loyolla; Cacilda Guerra; Luiz Vicente Vieira Filho; Marcos Domingos Agathão; Maria da Graça Mendonça Couto; **Diretor de Arte:** Adalberto Cornavaca; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistentes de Arte:** Miguel Luis Escámez Simón; Edemilson Ribeiro da Silva; **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; **Texto:** Carlos Eduardo Silveira Matos; Jussara Maturato; Paulo Brito; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Luiz Roberto de Godoy Vidal; Maria Clara Cescato; Maria Ester Menezes Martinho; Newton Roberval Eichemberg; Regina Amarante; **Pesquisa:** Stella Maria Quentel; **Assessoria Técnica:** Geraldo Coen; Gustavo José Ferreira Grubba; Lauro de Lauro; Pierluigi Piazz; Sérgio Jorge Batarce; Lucas Kenishi Shirahata. As fotos não creditadas pertencem à obra original. Seguem os créditos das demais: Luiz Antônio Ribeiro/ISTO É. © APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** Editora Rio Gráfica Ltda. e AM Produções Gráficas Ltda. **Impressão:** JBIG.

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Lda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.



NOVA MÚSICA ELETRÔNICA

No começo, a música eletrônica produzia apenas sons muito simples, com osciladores discretos. Hoje, a tecnologia digital permite complexas matrizes sonoras: de pingos de chuva à imponência de uma orquestra.

A imaginação não tem limites. E, entre os que reafirmam essa assertiva todos os dias, os músicos aparecem com destaque. Sua capacidade de combinar sons e emocionar é tão poderosa que consegue nos levar ao riso ou às lágrimas. E para fazer isso eles vêm usando, ao longo do tempo, todo tipo de instrumento, dos tradicionais aos eletrônicos.

Entre os últimos, o oscilador é considerado fundamental. Sua construção baseia-se no fato de que, quando um objeto vibra, o ar em torno dele se expande e se contrai com muita rapidez, criando uma onda que o ouvido e o cérebro interpretam como som. Da mesma forma, quando aplicamos uma corrente elétrica a um fio condutor por meio de um modulador (uma bobina, por exemplo), o metal vibra, produzindo a mais simples forma de onda: a senóide.

Essa pequena unidade geradora de som é conhecida como um oscilador. O controle por variação de tensão foi, desse modo, durante décadas, o principal método usado na produção de música sintetizada.

Há vários anos, os estúdios de gravação têm inúmeros e diferentes equipamentos de processamento sonoro, e não raro um conjunto de unidades de filtragem e reverberação é tido como elemento essencial de um estúdio. Do mesmo modo, os tecladistas dos anos 70 costumavam apresentar-se completamente cercados por sintetizadores, cada qual com uma infinidade de controles. Ao considerarmos o que acontece num estúdio de gravação bem equipado, seria útil lembrar um jogo chamado cadeia-de-mensageiros. Nesse jogo, uma frase é passada de pessoa a pessoa e a última informa o que ouviu. Verifica-se então que uma frase clara e direta no início acabou transformada numa porção de palavras absurdas. Num estúdio de gravação, ocorre algo semelhante: a frase original seria um conjunto de sons e, ao invés da corrente de pessoas que ouvem e reproduzem versões deturpadas da original, tem-se um grupo de unidades de processamento sonoro, cada uma controlável e capaz de cumprir tarefas específicas.

Qualquer pessoa que tenha de trabalhar com esse equipamento provavelmente usará uma mesa de controle para fazer as conexões entre as



unidades ou as ligará diretamente. Os controles de cada uma são geralmente calibrados à mão por um engenheiro de som — e podemos imaginar os problemas de sincronização e comunicação que isso tudo implica.

Tecnologia artística

Foi para gente como Egberto Gismonti, que não pára de fazer experiências com sons, que se empregou a tecnologia digital em instrumentos.



Controle central

Esta interface conecta microcomputadores a sintetizadores digitais: é a unidade processadora MIDI MP401, da Roland. Ela controla todas as funções externas de sincronização, entrada e saída de gravadores, além da saída de sinais para o sintetizador e o computador rítmico. Significa que o computador ao qual se liga responde apenas pelo envio e recepção de instruções e pelo gerenciamento do espaço na memória. Esta interface foi produzida inicialmente para os micros Apple II e IIe, IBM PC e para o Commodore 64.

Na década de 70, os tecladistas enfrentavam outros problemas. A dificuldade imediata não era produzir uma seqüência de sons de cada instrumento — para isso, bastava o músico mover a mão de um teclado para outro. Também não havia problemas para tocar dois teclados ao mesmo tempo. E, para ajudar, a maior parte do trabalho em estúdio não era feita de uma vez: os sintetizadores eram gravados separadamente e cada parte sobreposta à anterior, por meio dos diferentes canais de um gravador. Contudo, a evolução mais significativa ocorreu nas unidades de geração de som dos sintetizadores e nas técnicas instrumentais em geral. Bom exemplo

disso foi o aparecimento do sintetizador de baixos controlado pelo teclado.

Nessa época, os baixistas de funk atingiram um nível tal que rivalizavam com os guitarristas. No fim da década, no auge desse estilo, surgiram os baixos sintetizados em teclados. Para o tecladista, isso trouxe novos problemas. Ele teria de assumir também as funções de baixista, trabalhando com o baterista para “amarrar” o ritmo e deixar de fazer música “de teclado”. E, à medida que novos sintetizadores analógicos foram surgindo no mercado, outros tipos de sons — trompete, saxofone e bateria — tornaram-se disponíveis. Cada vez mais os tecladistas adotavam um dispositivo simples para lidar com novas responsabilidades: o seqüenciador.

Trata-se de um dispositivo que coordena linhas melódicas (ou seqüenciais), utilizando chaves analógicas (potenciômetros) ativadas por um controlador de tempo (clock). Este, por sua vez, comanda o oscilador na produção de tons de frequências diferentes. Os seqüenciadores determinam também a duração — curta ou longa — dos sons, por meio do controle de portas ou do tempo de duração do chaveamento. Com isso, podem-se criar padrões de ritmos e compor estruturas melódicas. O objetivo do seqüenciamento é garantir que o intervalo ocorra no mesmo instante, em cada repetição da seqüência.

No final dos anos 70, poucos sintetizadores possuíam recursos completos de seqüenciamento, mas os músicos logo começaram a usar tudo o que havia de disponível. A música de discoteca produzida por Giorgio Moroder e Donna Summer é, em grande parte, de seqüenciador — especificamente do Micro Composer MC-4, da Roland. E artistas brasileiros de vanguarda, co-

Criando um ritmo

Pode-se produzir som eletrônico de duas maneiras: programando osciladores, filtros, geradores de envoltório e de ruídos — muitas vezes usam-se ruídos “brancos” para gerar sons em micros — e também captando sons reais por meio de um dispositivo de conversão de analógico para digital e vice-

versa. Ou seja, faz-se uma amostragem dos sons, que serve para criar uma espécie de gabarito. A partir deste, pode-se montar um modelo de onda sonora dificilmente conseguida por outros meios. Tente sintetizar o ruído de uma caixa de bateria em seu micro para ver a dificuldade.

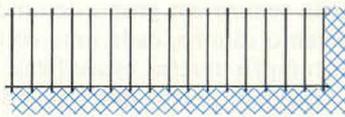


Analogico
A saída de um microfone é apenas uma corrente com tensão variável, análoga ao som que a produziu.



Digital
Um conversor digitaliza a onda numa série de valores de voltagem. Quanto maior o número de valores obtidos, mais preciso será o modelo digital.

CAIXA DE BATERIA



MEMÓRIA DA AMOSTRAGEM

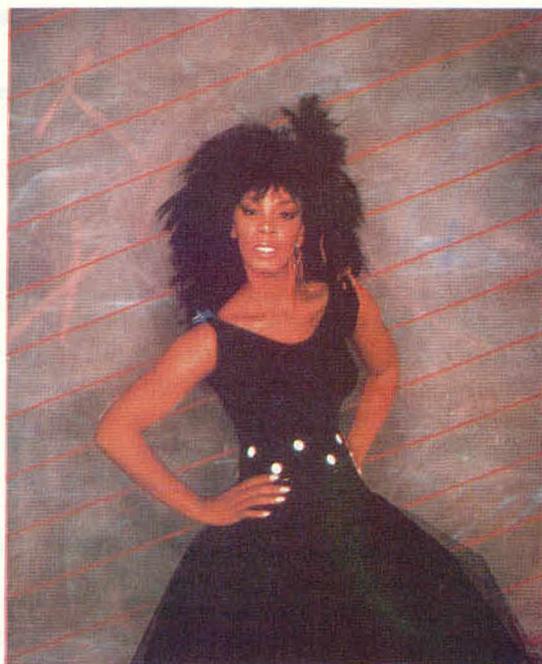


mo Egberto Gismonti, usando sintetizadores, desenvolveram um estilo completamente novo para se ajustar aos novos equipamentos. Os novos seqüenciadores digitais podiam controlar seqüências inteiras, para serem iniciadas ou interrompidas pela mudança de um controle. Durante a seqüência, o músico podia se afastar do teclado, dançar ao ritmo da música e então voltar a outro teclado.

Quando surgiram os sintetizadores digitais, os músicos descobriram que suas habilidades para usar seqüenciadores poderiam ser desenvolvidas ainda mais. Foi essa área do controle digital que despertou mais interesse em todos os músicos. Isso ficou demonstrado pelo sucesso do computador rítmico Linn, uma das primeiras unidades a utilizar o som amostrado — no caso, do baterista americano Steve Gadd. No Linn, gravaram-se os padrões de percussão em forma digital, do mesmo modo que se gravam os dados computacionais em discos. As seqüências resultantes de 1 e 0 são então codificadas em chips de ROM. A grande vantagem em se digitalizar o som está no resultado, que pode ser alterado em termos de tempo, ritmo, volume etc.

O estúdio de gravação e o tecladista de palco que até agora usamos como exemplo têm várias características em comum. Além de gravar música, o trabalho de estúdio reflete o crescimento na área do vídeo ocorrido nos últimos anos, trazendo um novo estilo e uma nova consciência do trabalho com imagens. Os produtores exigem que a música de cada videoclip contenha esse aspecto.

Se a música de um videoclip for produzida com instrumentos tradicionais, a sincronização com a imagem será feita pelo mesmo processo



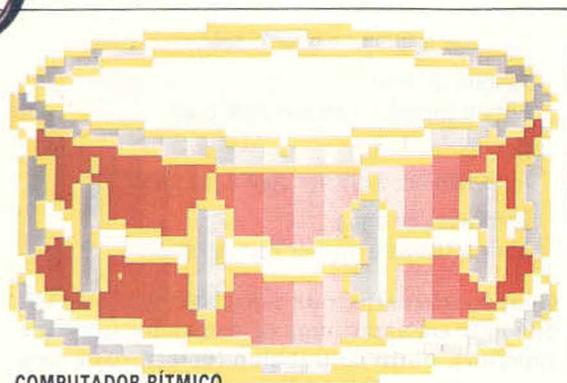
Dupla dinâmica

Donna Summer gravou um dos primeiros discos com ritmo produzido por sintetizadores. A idéia e o trabalho foram do produtor Giorgio Moroder.

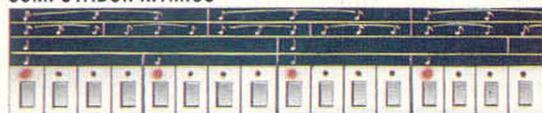
usado em cinema. Contudo, se vier de sons individuais e seqüências produzidas por sintetizadores digitais, muita coisa poderá dar errado. Vamos imaginar que, num certo videoclip, um vaso caia no chão e se quebre. O músico que está trabalhando na trilha produziu uma seqüência que se acelera até o ponto em que o vaso cai, e programou um acorde percussivo para o momento em que chega ao solo. A gravação começa, mas logo fica claro que a aceleração foi mal calculada, e a seqüência termina enquanto o vaso ainda está na mesa. O músico tenta então o acorde de percussão, que está gravado em outro canal. A gravação recomeça e, dessa vez, o acorde é gravado com 1 microssegundo de atraso. Os músicos necessitam, portanto, de uma forma de interligar os instrumentos digitais de modo que tudo aconteça na hora certa.

O tecladista tem problemas similares nas apresentações ao vivo. Seu equipamento pode incluir, digamos, dois sintetizadores digitais de marcas diferentes, além do Linn. Ele tem seqüências gravadas em todos, mas, como elas não estão no mesmo compasso, geralmente acaba colocando o Linn no automático e tocando a outra seqüência com as mãos. O resultado é que seu segundo sintetizador permanece inativo. Esse músico precisa de um meio de interligar seus instrumentos, para que todo o material seqüenciado seja acionado no tempo certo. E ele também necessita que o seqüenciador de seu primeiro sintetizador toque os sons pré-programados do segundo. Além disso, qualquer equipamento que utilize deve se adaptar a outros sintetizadores, além dos seus.

No próximo artigo, examinaremos a solução para isso tudo: a interface MIDI. Lançada em 1983, a partir de um acordo estabelecido entre os grandes fabricantes de hardware, essa unidade foi projetada para permitir que um determinado sistema digital (como um computador) controle outro (um sintetizador, por exemplo).



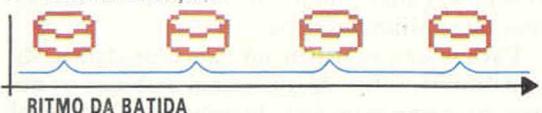
COMPUTADOR RÍTMICO



Teclas programadas

Um ritmo padronizado pode ser composto com um computador rítmico (Linn LM1, Roland TR-707, Oberheim DX), por meio do teclado. Na execução, o

dispositivo verifica quais as teclas programadas, e reproduz a batida no tempo certo — o qual se pode mudar a qualquer instante.



A CHAVE DOS DADOS

Sistemas gerenciadores de bancos de dados organizam a informação em registros contendo campos de dados de vários tipos. Mas, para acessar determinado registro, devemos considerar o conceito de "chave".

Instrumento incorporado a um SGBD (Sistema Gerenciador de Banco de Dados), a chave ajuda a localizar registros específicos. Vamos considerar um manual de manutenção de automóvel. Quando queremos regular o carburador, sabemos que a informação sobre como proceder está em algum lugar do manual. Como pode estar na página 36, esse número passa a ser considerado a chave para a informação. Mas talvez não saibamos que o capítulo sobre essa peça se encontra na página 36; por isso, recorreremos ao índice. Este dará o número da página, permitindo-nos ir direto ao ponto, sem necessidade de folhear cada uma das páginas.

É isso que ocorre com um SGBD. Cada registro num arquivo de banco de dados possui um único número de identificação (a chave primária). Mas, para encontrar determinado registro, você tem de escolher entre examinar um por um, seqüencialmente, até obter o dado procurado, ou, se já conhece o número do registro, ir diretamente a ele. Você também dispõe da opção de usar um dos campos como chave (no caso, uma chave secundária). Se tivermos um banco de dados sobre manutenção de automóveis, deveremos utilizar o campo NOME-PEÇA como chave.

A maioria dos SGBDs permite designar campos específicos como chaves. Quando um campo (NOME-PEÇA, no caso) for designado como campo-chave, o SGBD manterá uma tabela interna de palavras relacionadas com o campo especificado, junto com o número do registro apropriado (chave primária).

Quando você quer o registro que trata de carburadores, o SGBD procura na tabela NOME-PEÇA até encontrar CARBURADOR, verifica o número de registro correspondente e o acessa. Eis como se deve implementar tal esquema num SGBD simples escrito em BASIC:

```
INPUT 'ENTRAR COM CAMPO-CHAVE';CHA$
INPUT 'ENTRAR COM PALAVRA A
  PROCURAR';PAL$
GOSUB 20000 : REM SUB-ROTINA DE PESQUISA
PRINT REGISTRO
```

Uma entre várias matrizes foi selecionada, pelo

uso de CHA\$, e pesquisada por uma sub-rotina que usa o string PAL\$ como chave de pesquisa. Se a sub-rotina de pesquisa for suficientemente poderosa, terá condições de localizar um registro mesmo que se cometam pequenos erros de digitação na chave. Uma rotina simples como essa não depende da existência de uma tabela de chaves. Técnicas de pesquisa direta nos registros serão suficientes.

Os primeiros SGBDs para microcomputadores eram simples programas que qualquer amador poderia escrever em BASIC. Hoje, os micros de 16 bits contam com poderosos bancos de dados, que até há pouco estavam além do alcance dos micros mais simples. Mas um grande impulso foi dado com o advento do Sinclair QL, com seu processador 68000. A Psion (fabricante inglesa de software), utilizando os poderosos minicomputadores VAX, desenvolveu, para o Sinclair QL, o banco de dados Archive, pondo ao alcance do usuário amador avançados recursos de manipulação de dados.

Como exemplo de SGBD, vamos criar um banco de dados trivial, com quatro registros, sobre manutenção de automóveis. Cada um deles consistirá em apenas dois campos, NOME-PEÇA e CONserto:

```
Carburador
Remover a tampa e girar o botão
Tanque de óleo
Abrir o tampão e encher com óleo
Bateria
Abrir tampas e encher com água destilada
Borrifador de água
Abrir a tampa e encher com água
```

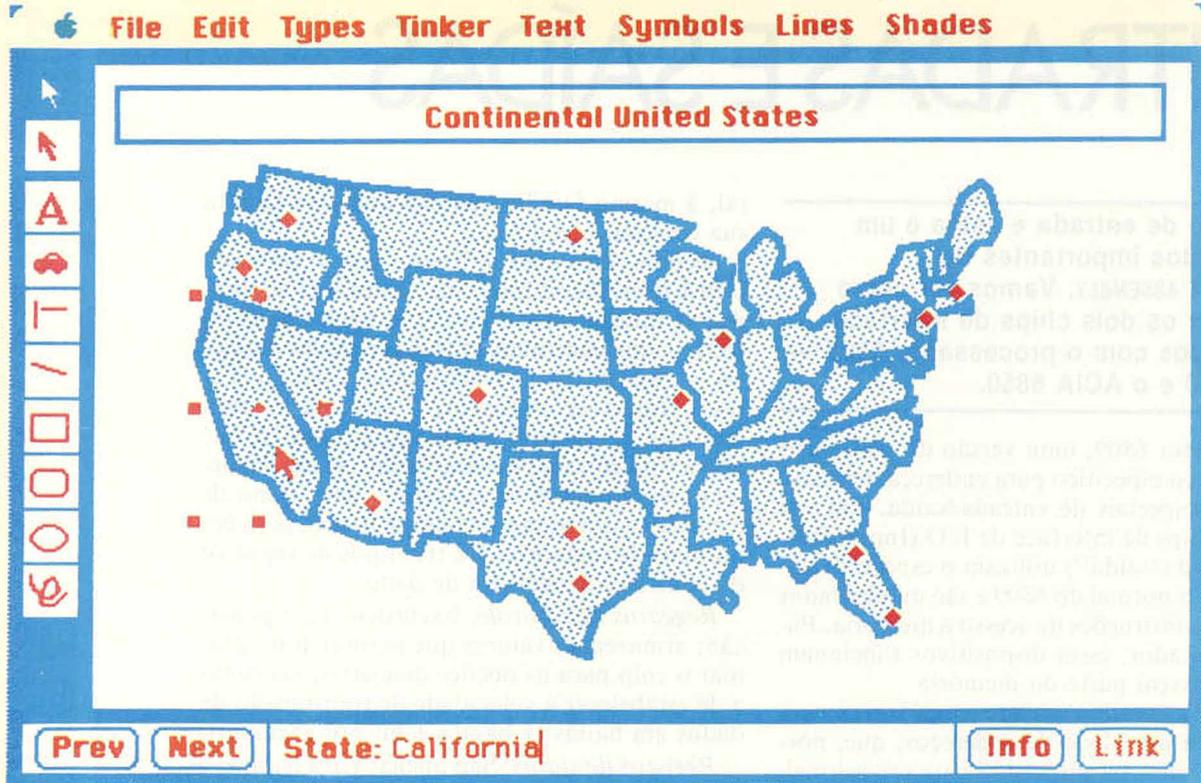
Quando o usuário roda o Archive, a tela do QL divide-se em áreas: uma de comandos (na parte superior), outra de trabalho (no centro) e uma terceira para exibição (na parte inferior).

Para criar um novo arquivo de banco de dados, seleciona-se o comando CREATE. No caso do banco de dados para manutenção de automóveis, digita-se CREATE 'CARRO' e entra-se com os nomes dos campos:

```
NOME-PEÇA$
CONserto
```

O sinal \$, com os nomes dos campos, indica serem estes alfanuméricos.

Para inserir registros no banco de dados, digite INSERT, a fim de que sejam exibidos os nomes dos campos na área de trabalho. Então você



pode entrar com os dados para cada campo. Pressionando a tecla [F5], um registro corretamente introduzido será acrescentado ao arquivo do banco de dados. Quando todos os registros que se quer introduzir no arquivo forem digitados, pode-se sair do modo INSERT acionando [Escape]. O encerramento do arquivo se faz pela digitação de CLOSE.

Para procurar um registro num arquivo, este deve inicialmente ser aberto. Pode-se fazer isso digitando o comando LOOK, que permite apenas examinar os registros, ou OPEN, que faculta pesquisas e alterações. Após ter aberto ou examinado um arquivo, você emite comandos simples, como FIRST (que exibe o primeiro registro no arquivo), LAST (o último), NEXT (o seguinte) ou BACK (o anterior).

Para pesquisar um registro, acrescenta-se um argumento a um comando — por exemplo, FIND'CARBURADOR'. Este buscará em todos os registros até localizar o que coincide com o string entre apóstrofes.

Operadores lógicos (E, OU e NAO) também podem ser usados, como no comando SEARCH NOME-PECA\$='CARBURADOR' AND CONSERTO='GIRAR'. Esse comando procurará no arquivo até encontrar um registro contendo o string CARBURADOR no campo NOME-PECA\$ e o string GIRAR no campo CONSERTO\$ do mesmo registro. Usa-se o operador OU a fim de que um registro seja localizado caso o string especificado para um campo ou o string especificado para o outro campo coincida com o nome procurado.

Em geral os registros entram num banco de dados fora de ordem, mas será necessário acessá-los e imprimi-los segundo alguma classificação.

Suponhamos, por exemplo, que você se propôs introduzir num banco de dados todos os livros de uma biblioteca, por autor, título, editora e o padrão internacional para numeração de livros (ISBN, International Standard Book Number). A maneira mais simples de fazer isso é percorrer as prateleiras e introduzir os dados de cada livro encontrado. Mas, utilizando a capacidade de classificação inerente a qualquer SGBD, você pode ordenar os registros alfabeticamente (por autor, título etc.), ou por número (pelo ISBN).

É possível combinar uma classificação primária com outra, secundária. Suponhamos que você queira o arquivo impresso em ordem alfabética por autor. Cada registro será depois ordenado alfabeticamente pelos títulos das obras. Imagine uma lista muito maior, mas com estas características:

A herdeira, Sidney Sheldon
Memórias do cárcere, Graciliano Ramos
Infância, Graciliano Ramos
Ainda estamos vivos, J. M. Simmel
Se houver amanhã, Sidney Sheldon
Até o mais amargo fim, J. M. Simmel
São Bernardo, Graciliano Ramos

Ela pode ser ordenada por autor e, a seguir, pelos títulos mediante este comando:

```
ORDER AUTORS;A,TITULOS;A
```

O A, nesse exemplo, significa "lista por ordem alfabética ascendente".

Os SGBDs transformam em trabalho leve tarefas que em geral exigiriam muito tempo, como localizar registros e reordená-los de maneira específica.

Os EUA na ponta dos dedos

O banco de dados Filevision destinava-se originalmente aos computadores Apple II e III. Mas teve de esperar até que o maior poder de processamento do Macintosh permitisse que suas notáveis e únicas características fossem plenamente apreciadas. O programa utiliza um mouse para acessar o banco de dados. Junto com o programa há um sofisticado pacote para desenho, que permite criar uma estrutura gráfica para o banco de dados. As formas e os símbolos que você desenhar são relacionados aos campos do banco de dados. A figura corresponde à tela principal de um banco de dados de informações sobre os Estados Unidos. Se apontarmos o mouse para a Califórnia, um clique simples dará uma breve lista de dados acerca desse Estado; um clique duplo abre um campo de texto que armazenará maior quantidade de informações. Os dados registrados podem ser relacionados e classificados da mesma maneira que em qualquer banco de dados. Por exemplo: o programa permite especificar um critério de pesquisa — digamos, todos os Estados com população superior a 10 milhões de habitantes — e então exibir os resultados traçando esses Estados, no mapa, com contornos mais espessos.

ENTRADAS E SAÍDAS

O controle de entrada e saída é um dos aspectos importantes da linguagem ASSEMBLY. Vamos ver como funcionam os dois chips de interface mais usados com o processador 6809 — o PIA 6820 e o ACIA 6850.

O processador 6809, uma versão do 6502, não possui espaço específico para endereçamento ou instruções especiais de entrada/saída. Em vez disso, os chips de interface de I/O (Input/Output, “entrada/saída”) utilizam o espaço de endereçamento normal do 6809 e são manipulados por meio de instruções de acesso à memória. Para o processador, esses dispositivos funcionam como se fossem parte da memória.

Apesar de sua simplicidade e rapidez, esse recurso ocupa um bloco de endereços, que, portanto, deixa de estar disponível para uso normal. Em consequência, embora o 6809 possua um bus de endereçamento de 16 bits capaz de endereçar 64 Kbytes de memória, fica restrito a cerca de 56 Kbytes, excluída a memória para controlar o hardware e o software. Esses chips de interface possuem uma complexidade tão grande quanto a do próprio processador e pertencem, em ge-

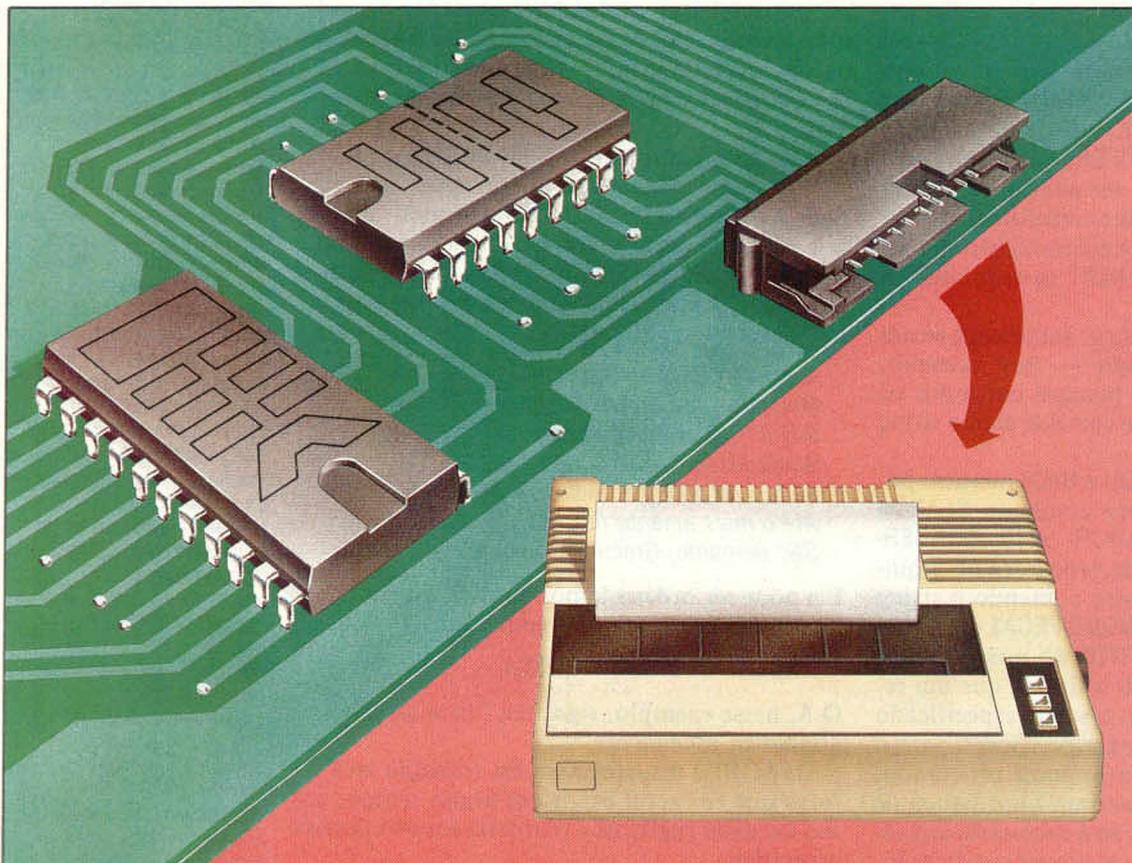
ral, à mesma família que ele, pois isso facilita sua conexão e controle.

Os dois chips mais utilizados com o 6809 são o PIA (Peripheral Interface Adaptor, “adaptador de interface periférica”), modelos 6802 ou 6821 — que controlam entradas e saídas paralelas —, e o ACIA (Asynchronous Communications Interface Adaptor, “adaptador para interface de comunicações assíncronas”), que opera com entradas e saídas seriais. Cada um deles possui numerosos registros e, para efeito de controle, basta tratá-los como se fossem posições normais da memória. Há três tipos de registro: de controle, de status e de dados.

Registros de controle. Exclusivos para gravação; armazenam valores que permitem programar o chip para as opções desejadas, tal como a de estabelecer a velocidade de transmissão de dados em bauds (1 baud = 1 bit por segundo).

Registro de status. São apenas para leitura, e seus valores indicam o status (estado) do chip. Esses valores mostrarão, por exemplo, se uma entrada foi recebida, se a última saída foi transmitida ou se houve algum erro.

Registros de dados. Contêm os dados que entram ou que saem e, portanto, são usados para leitura e gravação, simultâneas ou separadas.



Questões periféricas

As impressoras somente aceitam dados que estejam no formato apropriado e na velocidade certa. Seria um desperdício a CPU se ocupar com tais questões relativamente triviais. Por isso, ela envia os dados ao adaptador de interface periférica (PIA), que se dedica apenas à comunicação com a impressora.



A fim de economizar espaço na memória, costuma-se gravar mais de um registro no mesmo endereço — por exemplo, um registro de status e um de controle. O que nele aparece depende de se estar lendo ou gravando. De modo análogo, um registro de dados de entrada e outro de dados de saída também podem compartilhar um mesmo endereço.

O PIA 6820 dispõe de seis registros e ocupa 4 bytes consecutivos no espaço da memória. O chip contém, de fato, duas portas (A e B) independentes, cada uma utilizando três registros. O lado do chip ligado aos periféricos possui oito linhas de dados e duas de controle para cada porta. Podem-se conectar as linhas de controle a outras, também de controle, no periférico, de tal modo que determinem o status. A linha 1 destina-se apenas a sinais de controle de entrada, mas a de número 2 é programável para receber ou enviar sinais de controle. Em cada porta temos:

- Registro de dados, que funciona tanto para entrada quanto para saída, uma vez que se pode acessar cada bit de modo independente.

- Registro de endereçamento, onde se usa cada bit para acessar o bit correspondente no registro de dados como entrada (0) ou saída (1).

- Registro combinado de controle/status, no qual os registros anteriores compartilham o mesmo endereço. O estado de um dos bits no registro de controle determina qual deles aparece nesse endereço. A tabela à esquerda dá o desvio do endereço acessado pelo chip para cada um dos registros.

Os bits no registro de controle/status são determinados pela tabela a seguir:

Bit	Função
7	Bit de status para a linha de controle 1; ativado com valor 1 ao receber um sinal de controle e automaticamente zerado quando se lê o registro de dados
6	Bit de status para linha de controle 2; como o bit 7
5	Determina o uso da linha de controle 2 para entrada (0) ou para saída (1)
4	Determina a natureza do sinal de controle na linha 2
3	Com a opção para entrada na linha de controle 2, o valor 1 nesse bit ativa a interrupção do bit 6; se a linha for ajustada para saída, esse bit ajudará a determinar a natureza do sinal
2	Seleciona entre registros de dados (1) e de direção de dados (0)
1	Determina a natureza do sinal de controle na linha 1
0	Um valor 1 nesse bit ativa a interrupção do bit 7

O primeiro dos nossos programas-exemplo inicializa e usa um chip 6820 para controlar uma impressora por meio da interface padrão Centronics. Essa interface especifica um grande número de linhas de controle, bem como as linhas de dados. Uma linha de controle, chamada “strobe”, avisa a impressora que um caractere está a caminho. Essa linha conecta-se à linha de controle 2, que deve ser ajustada para saída. Outro sinal de controle da impressora, denominado “acknowledge”, indica que ela está pronta para receber o caractere seguinte. Este deve ser conectado à linha de controle 1. As oito linhas

de dados conectam-se às oito saídas de dados da porta PIA.

Para instalar a porta, devemos selecionar o registro de endereçamento dos dados e programar para saída todos os 8 bits. Depois, selecionamos o registro de dados e ajustamos a linha de controle 2 para saída. Para usar o chip, lemos de modo contínuo o registro de controle/status até um dígito 1 aparecer no bit 7, indicando que a impressora está pronta para um caractere. Podemos então gravar um caractere no registro de dados, que envia um sinal à linha de controle 2. O bit 6 assumirá o valor 1 quando o caractere for transmitido. Lemos, então, o registro de dados para limpar os bits 6 e 7 e repetir o processo até o último caractere. O processo de envio e recepção de sinais de controle entre o processador e os periféricos é conhecido por “handshaking”.

Devemos supor que o endereço de base do PIA seja mostrado na tabela de endereços localizada na posição \$3000. Na entrada da sub-rotina para a impressora, o registro A do processador contém o índice dessa tabela, e o registro Y, o endereço do string a ser impresso. Este é armazenado no formato normal, ou seja, com o byte indicador de tamanho em primeiro lugar. Há duas sub-rotinas, uma para preparar a porta e outra para imprimir o string.

ACIA 6850

O chip ACIA 6850 é um UART (Universal Asynchronous Receiver/Transmitter, “receptor/transmissor assíncrono universal”), utilizado para comunicação serial, que emprega o protocolo RS232 e, eventualmente, um modem. Possui quatro registros e ocupa dois endereços. Há cinco conexões com o chip, no lado dos periféricos: uma linha para os dados transmitidos, outra para os recebidos e três linhas para troca de informações de controle. Duas delas são para sinais de controle de entrada (DCD — Data Carrier Detect, “detecção de transporte de dados”; e CTS — Clear To Send, “limpeza para transmissão”), e outra é para sinais de saída (RTS — Request To Send, “pedido de transmissão”). Podem-se conectar essas linhas às de nomes semelhantes num periférico padrão RS232.

Os registros ACIA são mostrados na tabela da margem esquerda. No de controle, o bit mais significativo (bit 7) serve para ativar as interrupções, possibilitando a recepção de dados. Os bits 5 e 6 ativam ou desativam interrupções na transmissão e determinam o tipo do sinal de controle enviado pela linha RTS. Os bits 2, 3 e 4 determinam o tamanho do “pacote” transmitido. No envio de um byte por uma porta serial, usam-se, em geral, pelo menos 10 bits. A transmissão começa com o bit inicial (start bit), que informa o envio de dados ao receptor. A unidade de dados transmitida possui 7 ou 8 bits e pode conter 1 bit de paridade. O bit extra de paridade auxilia a detecção dos erros de transmissão. Por fim, existem 1 ou 2 bits de parada (stop bit). As opções são:

Porta	Registro	Desvio
A	Dados A	0
	Direção de dados A	0
	Controle/Status A	1
B	Dados B	2
	Direção de dados B	2
	Controle/Status B	3

Registro	Desvio
Registro de controle	0
Registro de status	0
Transmissão de dados	1
Recepção de dados	2



Registro de controle			Número de bits de dados	Paridade	Número de bits de parada
Bit 4	Bit 3	Bit 2			
0	0	0	7	Par	2
0	0	1	7	Ímpar	2
0	1	0	7	Par	1
0	1	1	7	Ímpar	1
1	0	0	8	Nenhuma	2
1	0	1	8	Nenhuma	1
1	1	0	8	Par	1
1	1	1	8	Ímpar	1

Os dois bits menos significativos (0 e 1) determinam a velocidade de transmissão e recepção. Para isso, estabelece-se um divisor para o valor do clock. O 6850 não possui clock, devendo, portanto, ser complementado com um externo, ajustado em 1.760 Hz.

Registro de controle		Divisor do valor do clock
Bit 1	Bit 0	
0	0	1
0	1	16
1	0	64

A combinação que não é mostrada na tabela, quando ambos os bits são 1, provoca uma reinitialização geral no chip.

No registro de status, os bits possuem as seguintes funções:

Bit	Função
7	Interrupção requerida
6	Ativado com valor 1 ao ocorrer erro de paridade na recepção
5	Ativado com valor 1 quando há sobrecarga do receptor, isto é, se bits demais são recebidos e ocorre uma sobreposição de caracteres
4	Ativado com valor 1 quando há erro de formatação na recepção — número inadequado de bits de início e de parada
3	Ativado quando um sinal é recebido na linha CTS
2	Ativado quando um sinal é recebido na linha DCD
1	Ativado quando o registro de dados transmitidos está vazio
0	Ativado quando o registro de dados recebidos está cheio

Nosso segundo programa-exemplo utiliza um chip 6850 para receber, de um terminal, um string de caracteres, finalizado por um sinal de retorno do carro. Trata-se de programar adequadamente o chip e, então, verificar por meio de um loop se o registro de dados recebidos está cheio. Em caso afirmativo, removemos o byte de dados, que reinitializa o bit 0 no registro de status. Repete-se o processo até que o caractere recebido seja o de retorno do carro (13, no código ASCII). Devem-se ignorar quaisquer erros de transmissão, embora seja preciso checá-los. Para isso, mascara-se o conteúdo do registro de status para comprovar a ativação de algum bit indicador de erro. Utilizamos um protocolo bastante comum: 8 bits de dados, sem paridade, 2 bits de parada e um divisor de velocidade de clock (por 16). A primeira sub-rotina programa o chip, a segunda recebe os dados.

Programa PIA

```

TABLE EQU $3000
ORG $1000
Sub-rotina para inicializar a porta A

ASLA
Desloca A à esquerda para multiplicar por 2 (a tabela inclui endereços de 2 bytes)
LDX #TABLE
Obtém endereço de base do PIA
LDX A,X
CLR 1,X
Acessa registro de direção de dados
LDB #01111111
Dispõe todos os bits para saída
STB ,X
LDB #00101100
Desativa interrupções, ajusta linha de controle 2 para saída e seleciona registro de dados
STB 1,X
RTS

Sub-rotina para imprimir o string do endereço Y
ASLA
Desloca A à esq. para multiplicar por 2
LDX #TABLE
Obtém o endereço de base do PIA
LDX A,X
LDA Y+
Obtém o comprimento do string em A
LOOP1 BEQ FINISH
Verifica se o comprimento é zero
LOOP2 LDB 1,X
Verifica se está pronto o bit seguinte
ANDB #01000000
Mascara todos os bits, exceto o 7
BEQ LOOP2
Se não estiver pronto
LDB Y+
Obtém o caractere seguinte
STB ,X
Imprime-o
LOOP3 LDB 1,X
Verifica se foi transmitido
ANDB #01000000
Verifica o bit 6
BEQ LOOP3
Executa loop se não estiver pronto
LDB ,X
Lê o registro de dados para limpar os bits de status
DECA
Subtrai 1 do comprimento
BRA LOOP1
Obtém caractere seguinte
FINISH RTS

```

Programa ACIA

```

TABLE EQU $3000
ORG $1000
Sub-rotina para programar o 6850

Sub-rotina para inicializar o ACIA
ACIAST ASLA
Desloca A à esquerda para multiplicar por 2 (tabela de endereços de 2 bytes)
LDX #TABLE
Obtém o endereço de base do ACIA
LDX A,X
LDA #00000011
Reinicialização geral do ACIA
STA ,X
No registro de controle
LDA #00010001
Programa ACIA (8 bits de dados, sem paridade, 2 bits de interrupção)
STA ,X
RTS

Sub-rotina para aceitar string de caracteres
BUFFER EQU $4000
CR EQU 13
BSR ACIAST
Inicializa ACIA. O registro X contém o endereço do ACIA
LDY #BUFFER
Destino em Y
LOOP LDB ,X
Obtém o status
ASLB
Desloca o bit 7 do registro B para a carry flag do CCR
BCC LOOP
Volta atrás se esse bit não foi ativado — ainda não há pedido de interrupção
LDA 1,X
Obtém um byte de dados
STA Y+
Armazena-o
CMPA #CR
A é um retorno de carro?
BNE LOOP
Caractere seguinte
RTS

```



MÁQUINA DE DESENHO

As tartarugas, dispositivos robóticos relativamente baratos, ajudam os iniciantes a compreender os fundamentos da programação. Vamos ver os recursos de um versátil equipamento: o plotter Penman.

O BBC Micro está entre os computadores preferidos pelos pedagogos, podendo controlar inúmeros periféricos educativos — plotters, mouses, tartarugas. A grande vantagem do plotter Penman é reunir esses três dispositivos.

O conjunto compõe-se da unidade de controle, plotter móvel, fonte de energia, cabo de conexão RS232 e software correspondente. Quando fora de uso, a unidade de controle e o plotter ajustam-se, formando um conjunto compacto, de apenas 55 x 128 x 335 mm.

Para remover o plotter de seu invólucro, basta pressionar o grampo situado sob ele, retirá-lo e desenrolar o cabo conector paralelo. Ficam visíveis então três buracos, dentro dos quais se introduzem as canetas, e um orifício central, que acomoda uma outra caneta para desenhos do tipo executado com tartaruga. As canetas — hidrográficas — medem 40 mm; quando em posição, permanecem acima do papel por meio de molas. O comando para desenhar puxa a mola para baixo por ação de um eletroímã, e a ponta da caneta toca o papel.

Três rodas na parte de baixo movimentam o dispositivo. Duas delas, metálicas, respondem pela tração e a terceira, uma pequena roda de plástico que gira livre, apenas proporciona equilíbrio. As rodas metálicas são embutidas num invólucro áspero, que confere atrito adicional quando o aparelho se move sobre o papel. À frente de cada roda, um elemento fotossensor detecta a margem do papel ao registrar a diferença de brilho entre este e o material de baixo.

Dentro do plotter, três eletroímãs, conectados a pequenas alavancas, controlam, cada um, o movimento vertical de uma caneta. Ao lado, dois motores elétricos acionam independentemente as rodas. O plotter utiliza motores comuns, e não servomotores. Isso permite o desenho de curvas contínuas, embora exija um controle de software bem mais sofisticado, pois requer variação nas voltagens aplicadas aos motores, em vez de apenas uma série de pulsos.

Finalmente, um painel de circuito distribui a energia elétrica e decodifica os sinais provenientes da unidade de controle, enviando-os aos eletroímãs ou aos motores. Em cada eixo dos motores há um disco estroboscópico ladeado por dois diodos detectores de luz.



Esse disco gira em conjunto com o motor, gerando pulsos de luz nos diodos. Estes, por sua vez, informam à lógica interna do painel de circuitos a rotação do motor para que o computador possa calcular sua posição. Isso se mostra particularmente útil quando o Penman é usado como mouse, pois o computador precisa saber em que sentido e a que rotação as rodas estão girando, a fim de deslocar o cursor na tela.

O painel com o circuito impresso, dentro da unidade de controle, tem três chips principais. O primeiro, um microprocessador 6303 (desenvolvido a partir do Motorola 6800), possibilita a configuração de inúmeros sistemas operacionais e incorpora uma RAM de “rascunho” que o capacita a armazenar a posição do plotter. O segundo chip é uma interface periférica 6821; e

Personalidade dividida

O plotter Penman consiste num módulo de controle, conectado a um computador através de uma interface RS232C, e uma tartaruga. O cabo paralelo que liga a tartaruga ao módulo é bidirecional (ambos trocam informações). Enquanto a caixa de controle envia instruções à tartaruga para mover-se até determinada posição, esta, por sua vez, informa aos circuitos de comando qual a sua localização.



Interface RS232C

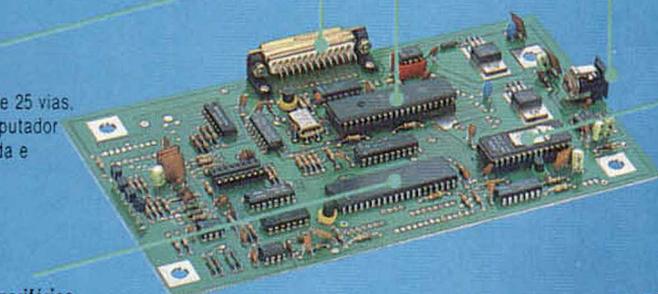
O conector D, de 25 vias, fornece ao computador sinais de entrada e saída.

Chip de interface periférica

O chip 6821 é muito usado para controlar a entrada e a saída de numerosos dispositivos digitais.

Cabo paralelo

Quando não está em uso, o cabo fica recolhido sob a chapa metálica da caixa de controle.



Alimentação

A fonte de energia do Penman é conectada aqui.

Processador 6303

Um processador 6303 permite ao plotter Penman configurar o sistema operacional.

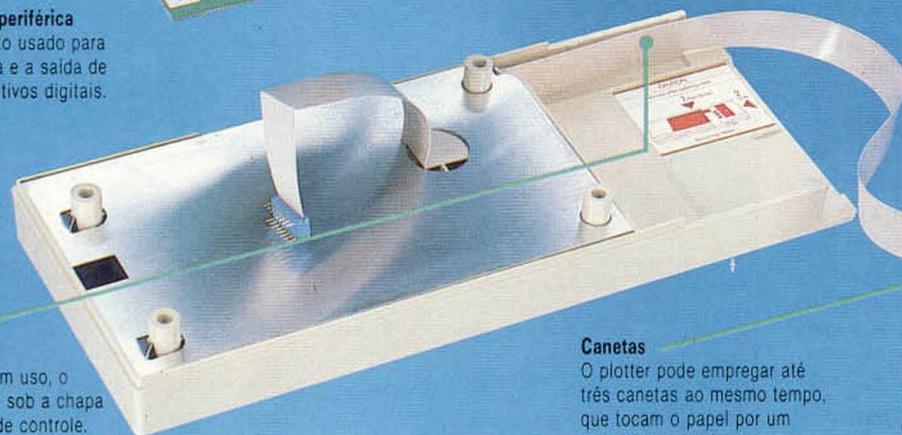
ROM do sistema operacional

Uma EPROM de 8 Kbytes contém o sistema operacional do Penman.



Solenóides

Controlam as alavancas que sustentam as canetas acima do papel.



Canetas

O plotter pode empregar até três canetas ao mesmo tempo, que tocam o papel por um mecanismo de alavanca.

Motores

O plotter é acionado por dois motores comuns, um para cada roda.

o terceiro, uma EPROM com os programas demonstrativos que o Penman executará se o cabo RS232C não estiver conectado.

Versatilidade operacional

O plotter Penman permite que o usuário escolha entre vários modos diferentes de utilização. No modo "emulador de terminal", o dispositivo é controlado por meio do teclado. Para selecionar esse modo, carrega-se o programa Pentlk diretamente ou, então, usando o robô Penman como mouse, a partir do menu principal. Carregado o software, enviam-se comandos às unidades de controle através da saída RS232C, sob a forma de códigos ASCII.

Para inicializar o Penman, digita-se PRINT 'I'. Ele receberá o sinal e determinará qual das três possíveis velocidades de transmissão, em bauds (bits/segundo), está sendo usada: 300, 1.200 ou 9.600. Feito isso, digita-se o comando H (home, "lar"), e tem início uma seqüência de movimentos que levam o robô ao canto inferior esquerdo da página.

De seu ponto de partida, o Penman tenta encontrar a margem da página usando seus fotosensores. Logo que a margem é encontrada, ele gira 90° e efetua a mesma ação ao longo da margem esquerda. Para garantir que o contraste entre o papel e a base seja captado pelo plotter, o conjunto inclui uma folha de papel preto, sobre a qual se apóia o papel de desenho. Depois que o Penman é colocado em "home", na margem inferior esquerda, ele estabelece sua posição a 5 cm de cada margem do papel (distância entre sua frente e o orifício central).

Executa-se a aplicação Pentlk no modo direto (um comando por vez) ou por intermédio de comandos reunidos em programas. Estes, por sua vez, são carregados (com LOAD), gravados em discos (com SAVE) ou executados (com RUN). O movimento no modo "emulador de terminal" baseia-se em coordenadas cartesianas — ou seja, o papel é considerado pelo robô como matriz.

Desse modo, quando recebe uma instrução para ir ao ponto (500,500), ele se move até essas coordenadas em vez de se mover quinhentos pas-



so em ambas as direções. Uma folha de papel A4 possui um máximo de 2.100 coordenadas no eixo x e 2.970 no eixo y.

Os comandos podem ser introduzidos nos modos absoluto ou relativo, conforme a instrução MOVE seja prefixada por um A ou por um R. Controlam-se as canetas por intermédio de comandos semelhantes aos da linguagem LOGO: P (pen) para selecionar uma delas, U (up) para levantá-la e D (down) para baixá-la.

O plotter Penman também executa, no modo direto, movimentos gráficos do tipo tartaruga. São um pouco mais complexos que os movimentos cartesianos, visto que a distância a ser percorrida deve entrar no micro em notação hexadecimal prefixada por um \$.

Isso porque os gráficos do tipo tartaruga se sobrepõem ao software normal que interpreta os movimentos cartesianos, indo diretamente aos bits nos endereços de controle, no interior do computador. Todavia, essa não é a única maneira de executar gráficos tipo tartaruga com o plotter Penman: o software aplicativo contém programas que permitem o controle do robô por meio da linguagem LOGO.

Controle em modo robótico

Um sistema semelhante para controle direto do Penman a partir do registro de dados do conector de expansões dirige o plotter em modo robótico.

Cada bit do registro controla um aspecto diferente dos movimentos do robô: os bits 0 e 1 e os bits 2 e 3 controlam, respectivamente, os motores direito e esquerdo; os bits 4 e 5 executam funções gerais dos motores, como ligá-los e desligá-los, e o bit 6 controla os movimentos da caneta, para cima e para baixo. O Penman dispõe ainda do recurso de transmitir ao registro de dados algumas informações a respeito de si mesmo, tais como erros de posicionamento e o aviso de que os fotossensores detectaram ou não a margem da página.

O plotter Penman aceita comandos de texto, que, tanto quanto suas aplicações, muito se assemelham aos utilizados nas impressoras/plotters disponíveis para grande número de microcomputadores pessoais. De fato, os caracteres de cada impressora apresentam notáveis semelhanças, podendo variar em tamanho de 1 a 127 mm de altura e ser impressos nos quatro sentidos. Um incremento útil do Penman, inexistente nas impressoras/plotters convencionais, é sua capacidade de inclinar o texto.

Instrumento pedagógico

O manual que acompanha o plotter apresenta uma compilação de todos os comandos disponíveis, com algumas explicações sobre a maneira como são implementados. Há também uma explanação extremamente minuciosa sobre a organização e o funcionamento do hardware. Talvez seja um tanto avançado para principiantes, podendo transmitir a idéia de que o plotter Penman se destina somente aos usuários que já dominam por completo o funcionamento de seus microcomputadores.

Em resumo, pode-se considerar o plotter Penman um avanço a mais no desenvolvimento de uma tartaruga/plotter, capaz de ser usada para finalidades múltiplas. Quando corretamente utilizado, sua resolução gráfica aproxima-se à dos plotters convencionais, encontrados nos estúdios de desenho comercial.

No entanto, o software existente impede que ele se torne um equipamento profissional. Ademais, seus programas objetivam, antes de tudo, a área educacional: espera-se que os usuários escrevam seus próprios programas para acionar o dispositivo. Isso os levará a aprender, pouco a pouco, os princípios do software robótico. Alguém que vise a finalidades comerciais certamente não se interessará por esses detalhes mais sutis da programação, exigindo, em vez disso, um aparelho que seja fácil de usar e resolva com eficiência seus problemas.

PLOTTER PENMAN

DIMENSÕES

335 x 128 x 55 mm.

INTERFACE

RS232C, que permite conexão a qualquer computador compatível com a RS232.

RESOLUÇÃO

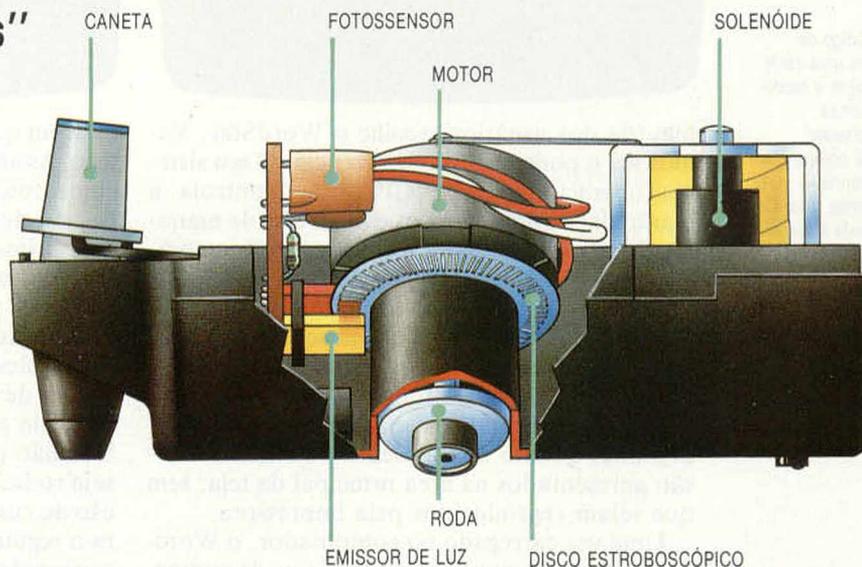
Tamanho do texto: 1 a 127 mm; resolução gráfica: 0,1 mm.

DOCUMENTAÇÃO

O manual fornece numerosos detalhes técnicos, permitindo aos usuários já avançados desenvolver todo o potencial do Penman. Mas um principiante pode sentir falta de maior número de informações em nível elementar.

O controle dos "passos"

Para que o sistema de controle "saiba" a posição do plotter, um mecanismo de retroalimentação no interior do robô informa o número de "passos" que foram dados. Trata-se de um disco de metal provido com numerosas fendas, ligado ao eixo de cada uma das rodas. Montado no painel do circuito, há um dispositivo que emite e recebe luz. À medida que o disco gira, produz-se um efeito estroboscópico entre o emissor e o receptor, na forma de uma série de pulsos. Pelo número de pulsos, os "passos" podem então ser calculados.



TEXTOS EM TELA

O processamento de texto, maior área de aplicações da microcomputação profissional, conquista atualmente os usuários domésticos. Quase todos escolhem o processador WordStar, recordista mundial de vendas.

O WordStar é um processador de texto orientado por menus, com formatação "transparente", ou seja, invisível para o usuário. A exemplo dos demais processadores, pode gravar documentos — desde uma única linha até páginas inteiras — em fitas ou discos e incorporá-los a outros textos. Pode localizar e substituir uma palavra ou expressão quantas vezes aparecer num documento, sendo capaz de alinhar o texto à esquerda, à direita ou de justificá-lo, isto é, variar o espaçamento entre as palavras para que todas as linhas tenham o mesmo tamanho.

Quase todos os softwares processadores de texto existentes no mercado possuem esses recursos. Contudo, na hora de adquirir um deles, a

maioria dos usuários escolhe o WordStar (dos), que permite a escolha da função desejada. Por meio dele, o usuário cria ou edita arquivos, na forma de documentos ou na de não documentos (usados, por exemplo, para conter as listas de nomes e endereços para mala direta); incorpora num único arquivo vários outros; seleciona a unidade de disco a ser utilizada; apaga, copia ou renomeia um arquivo; interrompe o programa e passa o controle ao sistema operacional; ou define o nível de auxílio necessário.

Nesse menu inicial, a escolha de cada opção é feita pelo acionamento de uma única tecla, dispensando-se o uso de [Control]. A tela mostra, então, o menu específico da função selecionada. A última das opções (definição do nível de auxílio) busca satisfazer tanto aos usuários novatos quanto aos experientes. Ela permite determinar como serão mostrados na tela os menus de comandos: quanto mais baixo o nível de auxílio requerido, menor o número de linhas reservadas para o menu e, conseqüentemente, maior o de linhas disponíveis para mostrar o texto. A maioria dos comandos é mnemônica — na me-

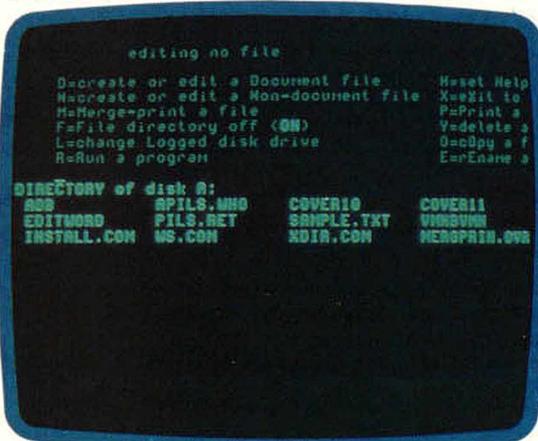
Menu inicial

Apresenta as funções básicas disponíveis. O usuário escolhe então entre criar ou editar um arquivo, seja do tipo documento — em que utiliza a justificação de linhas, a transferência de palavras para a linha seguinte e outros parâmetros de formatação padronizados —, seja do tipo não documento. Pode ainda copiar, renomear ou apagar um arquivo. Em resumo, essa parte do programa trata das tarefas de manutenção.

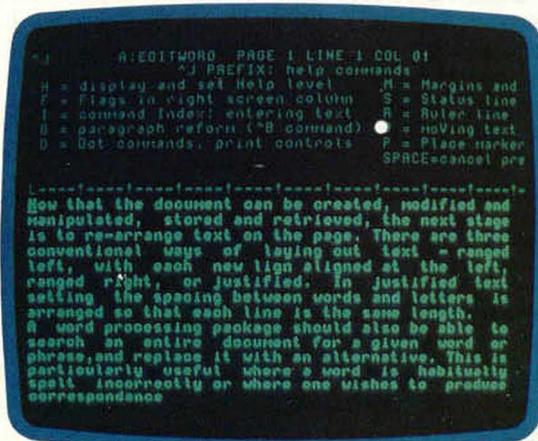
Menu de auxílio

Acessado pelo código de controle J, fornece uma série de explicações sobre o modo de operação de partes específicas do software. Acrescentando ao código J um caractere suplementar — H, B, F, D etc. —, obtemos uma explicação completa a respeito daquelas seções do WordStar.

Menu inicial



Menu de auxílio



maioria dos usuários escolhe o WordStar. Vamos ver o porquê dessa preferência. O seu sistema operacional — o CP/M — controla a manipulação dos arquivos e as tarefas de manutenção. Acionando-se a tecla [Control], o computador diferencia entre comandos e caracteres. O usuário pode determinar que o menu de opções permaneça exposto numa seção de dez linhas, reservada para essa finalidade na parte superior do vídeo. Os comandos que definem a forma com que o texto aparece — tanto ao ser digitado, quanto no momento da impressão — são apresentados na área principal da tela, sem que sejam reproduzidos pela impressora.

Uma vez carregado no computador, o WordStar apresenta um menu inicial (ou de coman-

dida em que o podem ser com apenas um caractere. Assim, a tecla [D] cria um arquivo de documentos, enquanto a tecla [N] cria um arquivo de não documentos.

A primeira informação solicitada pelo sistema é o nome do arquivo a ser criado. Depois, o sistema passa para o menu de operação. Este consiste num índice de funções, como, por exemplo, deslocamento do cursor, inserção ou eliminação de textos e lembretes dos códigos de controle para acessar outros menus. O WordStar não pressupõe que o computador no qual seja rodado disponha de setas para movimentação do cursor; o movimento de um caractere para o seguinte é definido pela tecla [Control], em conjunção com as teclas [E], [S], [D] e [X], que

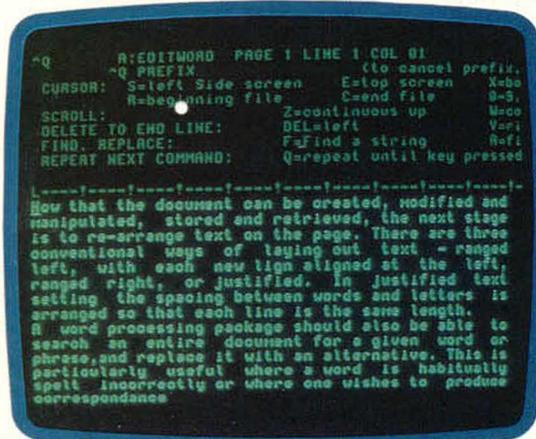
formam um “diamante” no teclado (para cima, para baixo, à esquerda, à direita). Portanto, os computadores com teclas de setas para controle de cursor têm essas funções duplicadas ao rodar o WordStar. Neste, esse controle é bastante versátil, regulando o seu deslocamento por caractere, palavra, linha ou parágrafo.

O mesmo princípio foi aplicado na função de eliminação. Caracteres isolados podem ser suprimidos, tanto no ponto em que está o cursor, quanto imediatamente à sua direita. Também é possível apagar palavras inteiras, à direita da posição do cursor, ou eliminar linhas.

As funções de controle também permitem deslocar o texto verticalmente na tela, sem mudanças no posicionamento do cursor. A utilidade desse recurso é evidente, caso seja preciso rever partes anteriores do texto.

Por último, o usuário tem a possibilidade de optar entre inserir um caractere na posição do cursor ou substituí-lo por outro, introduzido por meio do teclado. Essa função é do tipo liga/desliga, ou seja, o acionamento da tecla apropriada ([V], no caso) transforma o modo de inserção no de substituição. Este permanece ativado até que [V] seja pressionada de novo, juntamente com a tecla [Control]. O software controla automaticamente a paginação — pela qual se define o número de linhas por página — e, quando cada linha chega ao final, transfere as palavras para a seguinte.

Menu de comandos rápidos



Essas funções cobrem praticamente todos os aspectos da criação e edição de documentos. No entanto, o WordStar oferece recursos suplementares. Por meio deles é possível, por exemplo, centralizar títulos, sublinhar palavras e linhas, usar caracteres em negrito, e imprimir sobrescritos e subscritos (m^2 e H_2O , por exemplo). Todos esses recursos estão disponíveis no momento da criação do texto ou mais tarde, para se processarem as correções.

A função de “busca e substituição” é acessível por meio do menu Q (quick, “rápido”). Em primeiro lugar, o usuário digita os caracteres correspondentes à palavra ou frase a ser pesquisada. Depois, ele indica se deseja encontrar uma, várias, ou todas as ocorrências da palavra ou fra-

se digitada. Decide, a seguir, se elas devem ser apenas localizadas ou substituídas por outras palavras ou frases. Realiza-se a substituição sem qualquer instrução adicional.

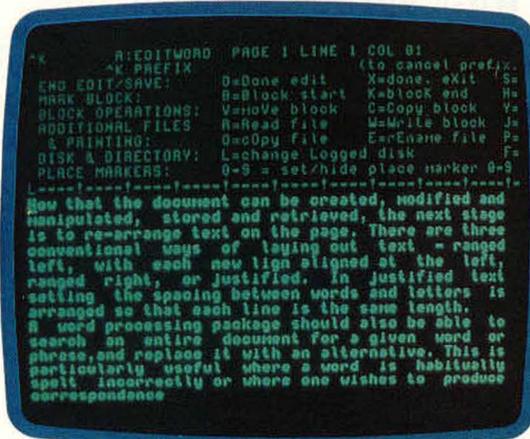
De modo semelhante, é possível copiar ou transferir blocos de texto de um lugar para outro, num documento, através das funções do menu K (block, “bloco”). Os três outros menus suplementares — J, O e P — abrangem, respectivamente, a explicação de códigos de controle, a formatação dos textos na tela e o modo em que estes devem ser impressos.

As características do WordStar permitem que ele seja executado numa grande variedade de computadores, desde que rodem no sistema operacional CP/M e, de preferência, aceitem 80 colunas na tela e letras minúsculas. Muitas dessas máquinas vêm equipadas com teclas de função programáveis. O IBM PC, por exemplo, tem dez delas. Cada uma pode ser programada para representar uma série de caracteres de controle, uma palavra ou uma frase empregada com mais frequência. Dessa maneira, fica ainda mais fácil a sua inserção no texto.

O WordStar no Brasil

Os usuários brasileiros do WordStar sempre se ressentiram das dificuldades da acentuação. Nessa mesma palavra, para se escrever a última sílaba, digitava-se a letra c, depois um código de superposição e uma vírgula; tudo isso para “fa-

Menu de blocos



zer” a cedilha. Escrevia-se então a letra a e repetia-se o código; a seguir vinham o til e a letra o. Todos os caracteres, das palavras e dos códigos, apareciam na tela.

Em 1984, porém, a BraSoft lançou uma versão “acentuada” do WordStar, na qual os caracteres aparecem na tela tal como serão impressos. O programa se destina a micros de 16 bits (IBM PC e compatíveis), podendo também empreender a separação silábica em português. Vem acompanhado de um corretor ortográfico, um programa que funciona acoplado ao WordStar e que abrange cerca de 20.000 verbetes em inglês. Os fabricantes empenham-se na tradução para o português desse “dicionário eletrônico”.

Menu de comandos rápidos

Permite ao usuário posicionar o cursor no início ou no fim da linha, da tela ou do documento, bem como utilizar as funções de busca e substituição. É acessado pelo código de controle Q, primeira letra da expressão inglesa quick command menu (“menu de comandos rápidos”).

Menu de blocos

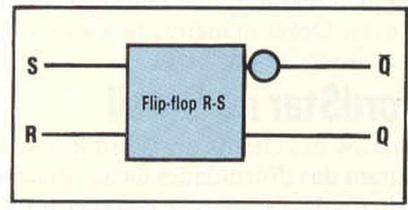
O menu K — denominado a partir da última letra da palavra block, “bloco” — controla a gravação e leitura de arquivos, o deslocamento e a cópia de blocos de texto. Ou seja, duplica algumas rotinas de manutenção do próprio software, acessíveis pelo menu inicial: gravação, eliminação, cópia e junção de arquivos.



CIRCUITOS FLIP-FLOP

Os circuitos seqüenciais são capazes de produzir um sinal de saída contínuo, em resposta a um só pulso de entrada. Nesta matéria examinaremos o desenho e a função de um circuito assim, o flip-flop R-S.

Há vários tipos de flip-flop, todos baseados no mesmo princípio. O flip-flop R-S possui duas linhas de entrada e duas de saída (Q e \bar{Q}):

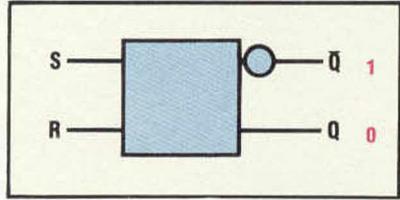


Desenha-se o circuito de forma que as linhas de saída estejam sempre em oposição:

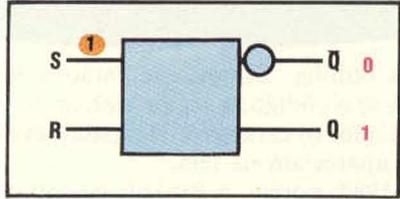
- se $Q = 1$, então $\bar{Q} = 0$ (estado SET);
- se $Q = 0$, então $\bar{Q} = 1$ (estado RESET).

Supondo que o flip-flop comece no estado RESET, um pulso na linha S fará com que o circuito "salte" (flip) para o estado SET.

1) Estado inicial (RESET)

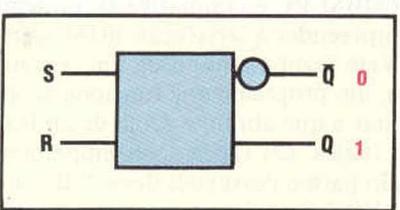


2) Um pulso na linha SET



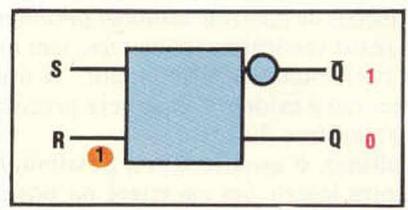
Quando cessa o pulso de entrada da linha S, o circuito permanece no estado estável SET.

3) O circuito permanece estável (SET)



Um pulso enviado por R faz o circuito saltar (flop) para seu estado RESET inicial.

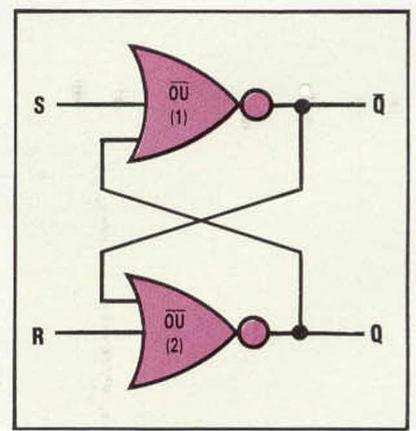
4) Um pulso na linha RESET



Descrita a função de um flip-flop R-S, passamos a examinar mais de perto os elementos lógicos do circuito.

O circuito flip-flop R-S

Podemos construir um flip-flop R-S mediante a ligação de duas portas E ou duas portas OU (caso deste exemplo), de modo tal que a saída de cada uma forneça uma das entradas para a outra. É esse "retorno em loop" dos sinais lógicos que permite a capacidade de "memória" do flip-flop:

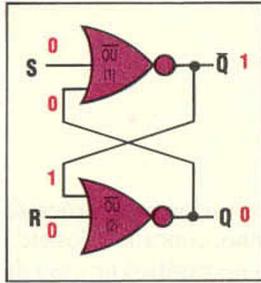


Vamos acompanhar as funções SET e RESET e ver como a combinação de portas \overline{OU} nos permite obtê-las. Supondo que o flip-flop inicie no estado RESET e que não haja pulsos de entrada, o estado do circuito será o estável (lembre-se de que a porta \overline{OU} somente fornecerá uma saída de 1 se ambas as entradas forem 0).

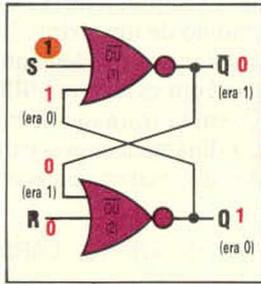
Um pulso na linha S desestabilizará essa disposição, alterando a saída "não Q" (\bar{Q}) para 0. Isso afeta a entrada que "retornou em loop" para a segunda porta \overline{OU} (2), fazendo com que a saída proveniente dessa porta (Q) se modifique para 1. Assim, se o pulso ainda está presente na primeira porta \overline{OU} (1), as entradas para essa porta são ambas 1, e a saída, 0: o circuito alcança um estado estável, isto é, SET.



1) Estado inicial (RESET)

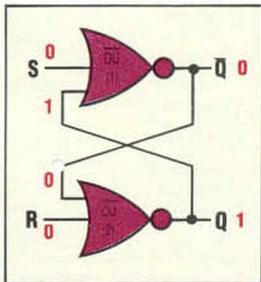


2) Um pulso na linha SET

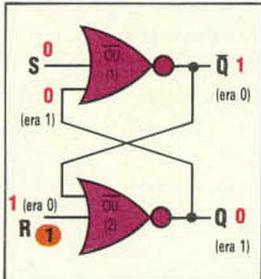


Mesmo quando se retira o pulso da linha S, o circuito permanece estável. Quando se envia um pulso pela linha R, coloca-se o circuito num estado não estável. Após um processo semelhante ao já descrito, o circuito volta a um estado estável RESET.

3) O circuito permanece estável (SET)

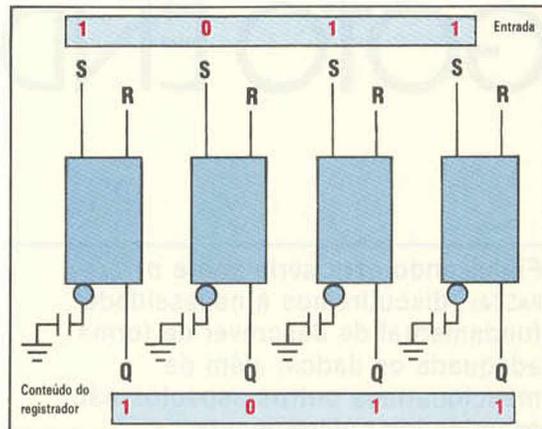


4) Um pulso na linha RESET



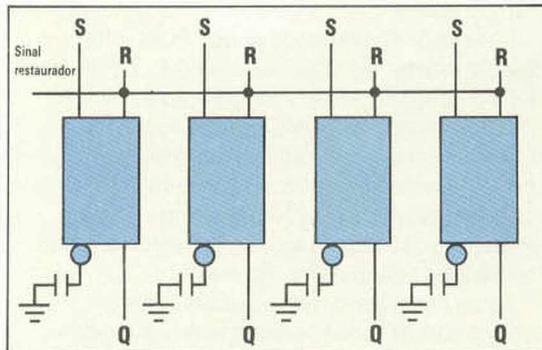
Registadores

O microprocessador é em grande parte formado por uma série de registadores, como os acumuladores, os indexadores e os que processam instruções. A maioria dos registadores manipula palavras de 8 bits (números binários no limite de 0 a 255). Como têm de aceitar e armazenar informações binárias, esses registadores são constituídos por uma série de oito flip-flops. Para que um registor de 4 bits armazene o número binário 1011, por exemplo, é necessário apenas fornecer o padrão binário às linhas S.



Observe que, nessa posição, a saída “não Q” não é utilizada. De acordo com o padrão binário aplicado às linhas S dos flip-flops, obtemos uma saída correspondente das linhas Q. Para substituir o primeiro número armazenado no registor por outro, por exemplo, 0110, não basta apresentar esse novo padrão binário para as linhas S. Com tal procedimento, o número resultante armazenado no registor seria 1111. Os 1 nas posições externas desse número são remanescentes do número anterior.

A solução será então restabelecer todos os flip-flops simultaneamente antes de armazenar o segundo número. Para tanto, convém conectá-los, permitindo restaurar o registor por um único sinal.



No próximo artigo da série continuaremos examinando circuitos seqüenciais, inclusive o flip-flop tipo D e o flip-flop J-K.

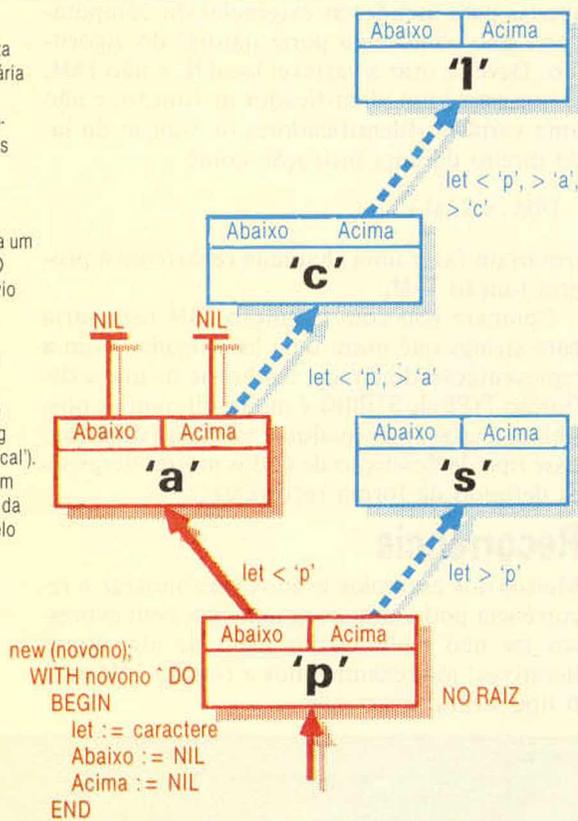
Exercício 7

- 1) Por que o circuito flip-flop é também chamado “biestável”?
- 2) Quando um computador é ligado, um flip-flop está neste estado:
 $Q = 0, \bar{Q} = 0, S = 0, R = 0$
 - a) Esse é um estado estável?
 - b) Se não for, para qual estado o flip-flop mudará?
 - c) Ele pode passar para um estado que não o da resposta anterior?
 - d) Ao ligar um computador, qual o processo para garantir que todos os registadores estejam num estado previsível?

Criando um novo nó

O programa TreeSorter utiliza uma estrutura de árvore binária para armazenar dados classificados por um campo-chave. Os itens são inseridos nos desvios inferior ou superior, conforme a ordem alfabética do campo. O procedimento INSERE cria um novo nó, enquanto DIRECAO determina a rota para o desvio NIL adequado.

O diagrama mostra uma árvore binária simples, onde cada nó mantém um único caractere (LET), e o modo de armazenamento de um string simples de minúsculas ('pascal'). A área em vermelho indica um estágio no desenvolvimento da árvore, tal como efetuado pelo procedimento INSERE.



```
FUNCTION TAM (S : STRING) : CARDINAL;  
BEGIN  
IF S = NIL  
THEN TAM := 0  
ELSE TAM := SUCC (TAM (S ^ PROX))  
END; |TAM|
```

O cabeçalho da lista é S, e a expressão S ^ PROX seleciona o campo indicador do registro seguinte da lista. Ou, então, ele pode ser considerado como uma lista que começa com o registro seguinte (PROXREG). Sempre que não encontramos um NIL, chamamos uma avaliação do tamanho de PROXREG e aumentamos com a função SUCC.

O exemplo anterior, bastante comum, permitirá escrever TAM sem o uso da recursividade. O código é maior e deve-se usar um contador local (como na versão da matriz) especificamente para evitar a recorrência.

```
FUNCTION TAM (S : STRING) : CARDINAL;  
VAR  
N : CARDINAL;  
BEGIN  
N := 0;  
WHILE S <> NIL DO  
BEGIN  
N := N + 1;  
S := S ^ PROX  
END;  
TAM := N  
END; |TAM|
```

Muitos compiladores PASCAL admitem "diretivas", que são instruções para o compilador — e não declarações ou comentários. A única diretiva realmente necessária pelo padrão ISO é FORWARD. Caso dois procedimentos precisem

chamar um ao outro, eles serão denominados "mutuamente recorrentes". Isso acontece muito raramente, mas coloca um problema: não podemos utilizar um objeto em PASCAL sem antes declará-lo ou defini-lo.

A solução será declarar somente o cabeçalho do subprograma, substituindo seu bloco pela diretiva FORWARD do compilador. Após a definição completa do outro módulo, o cabeçalho recebe sua forma abreviada (sem a lista de parâmetros), e o bloco é definido nesse ponto. Muitas vezes há outras diretivas para controlar as opções de controle durante a compilação, mas não deverão ser usadas com frequência, se você quiser garantir a portabilidade. Isso significa que, além de um símbolo especial (geralmente \$) aparecer como o primeiro caractere do comentário, as opções não portáteis podem não ser entendidas por um outro compilador.

Outras diferenças

Algumas implementações não padronizadas (especialmente a HiSoft) exigirão sintaxe um pouco diferente para as declarações indicadoras FORWARD. Há pouquíssimas "variações" desse tipo e nenhuma delas será encontrada em compiladores que seguem a definição ISO.

Uma descrição avançada de dados, muito importante no PASCAL e que não vimos, é a "variante". Para armazenar itens de diferentes tipos, utilizamos um registro com campos adequados. Mas, e se precisássemos tornar flexível a descrição de parte ou mesmo de todo o registro? De modo geral, poderíamos querer armazenar diferentes informações sobre as pessoas, dependendo, por exemplo, de serem ou não casadas. Um registro de variante possui sua "parte fixa" definida em primeiro lugar; depois, especifica-se a parte variável pela introdução de um seletor de variante (de qualquer tipo simples) e pelo uso das palavras reservadas CASE e OR. Por exemplo:

```
TYPE  
GENERO = (MASCULINO, FEMININO);  
VARIANTE = RECORD  
{CAMPOS COMUNS;  
CASE ESTCIV : BOOLEAN OF  
FALSE : ();  
TRUE : (DATACAS : STRING;  
CASE SEXO : GENERO OF  
MASCULINO : ();  
FEMININO : (ESPOSO : STRING))  
END; |VARIANTE|
```

Observe que as listas de campo vazias devem ainda estar entre parênteses. O espaço em um arquivo será fixo (de acordo com a maior variante), mas pode-se economizar memória com indicadores para variantes, como, por exemplo, NEW (P, TRUE, MASCULINO).

Os poucos pontos frágeis do PASCAL foram em grande parte eliminados pelos especialistas (e por Wirth, em MODULA-2). A linguagem é bastante poderosa (embora pequena), eficiente, destinada a usos gerais e fácil de aprender.



SORD

No competitivo mercado japonês, dominado por gigantescas corporações, a SORD, uma pequena empresa com algumas centenas de empregados, vem se destacando por seus inovadores projetos informáticos.

O curioso nome da SORD (combinação de Software e hardware) é adequado, já que ela sempre dedicou atenção tanto ao desenvolvimento do software quanto do hardware.

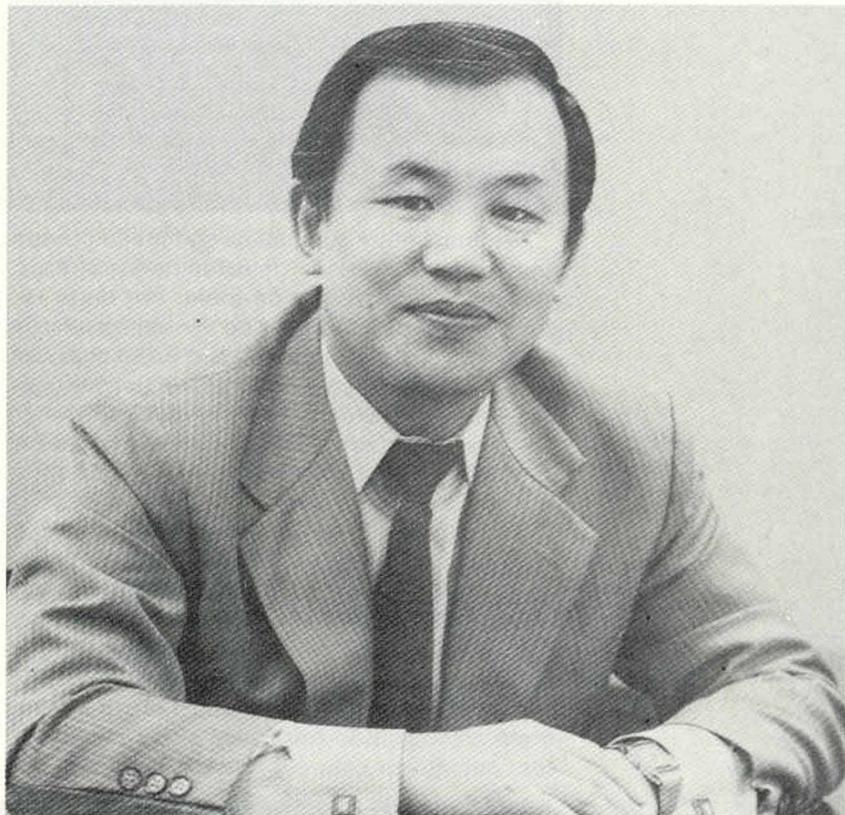
A companhia teve começo um pouco vacilante. Em 1967, seu futuro presidente, Takayoshii Shiina, deixou a universidade para reorganizar o departamento de marketing da Rikei, um conglomerado industrial de porte médio. O fato de Shiina ter conseguido esse emprego chega a ser algo incomum, visto que, no Japão, os empregados de uma empresa raramente mudam de emprego, nele permanecendo por toda a vida. Segundo consta, já em sua primeira entrevista, ele declarou que, como planejava dirigir sua própria firma, trabalharia na Rikei apenas por alguns anos.

Em 1970, ele e um amigo estavam prontos para lançar o novo empreendimento e constituíram a SORD com um pequeno capital. Shiina, no entanto, continuou na Rikei até dezembro daquele ano. O primeiro produto fabricado pela SORD foi um testador lógico de baixo custo; além disso, a empresa obteve também alguns contratos para trabalhos de programação.

Em 1971, a SORD começava a envolver-se numa ampla gama de negócios, mas o início da produção de equipamentos deu-se dois anos depois, e no final de 1974 a SORD apresentou uma unidade de discos flexíveis que operava com uma interface por ela desenvolvida. Seguiu-se, logo, o SMP-80/20, um dos primeiros computadores do Japão baseados no processador Intel 8080.

O SMP-80/20 foi um produto de grande sucesso e logo multiplicaram-se as encomendas. Em 1977, visando à expansão de sua empresa, Shiina vendeu à Toppan, uma das maiores gráficas do Japão, 20% das ações sob seu controle. Essa injeção de capital permitiu que a SORD criasse considerável base de software para sua lista cada vez maior de computadores. Desse trabalho resultou, em 1981, o PIPS (Pan Information Processing System, "sistema de processamento multiinformacional").

Um dos primeiros softwares integrados, o PIPS estava muito à frente dos pacotes existentes na época, e ajudou a consolidar a posição da



Takayoshii Shiina
Fundador e presidente da SORD.

SORD no mercado. Esse pacote combina as funções de planilha financeira, processador de textos e banco de dados, de tal modo que até pessoas que nunca manejaram um computador aprendem a usá-lo em poucas horas.

O sucesso fenomenal do PIPS no mercado interno japonês só pode ser entendido com uma explicação sobre suas características na época. No Japão, era prática corriqueira a venda de computadores sem software de apoio. Embora comum na Europa e nos Estados Unidos, o software aplicativo do tipo não profissional — para contabilidade, faturamento etc. — era virtualmente desconhecido no país.

A carência de aplicativos no mercado japonês deixava um espaço enorme para a SORD explorar. A empresa começou a desenvolver a série M200 de computadores baseados no Z80 e, mais tarde, a série M23. O M5 foi introduzido para atender o mercado de jogos domésticos. Em meados da década de 80, foi comercializada a série M68 — que incorporava tanto o micro processador Z80 como o Motorola 68000 — e os pequenos micros comerciais M243, além do computador M285, de 32 bits, que roda o software VAX-11 para aplicativos CAD (Computer-Aided Design, "projeto auxiliado por computador").



PIPS

Desenvolvido pela SORD com características únicas, este software de uso geral adapta-se à maioria das aplicações comerciais, da contabilidade aos gráficos promocionais. Demonstra a peculiar visão japonesa de marketing, pois só pode ser utilizado nos equipamentos SORD. Embora a eficiência do PIPS constitua um grande atrativo, sua especificidade limita bastante a demanda.

Nenhuma outra companhia no mundo produz uma linha tão variada de computadores, cada um com vasto software de apoio. Por outro lado, a SORD parece falhar ao ver seus produtos como "fins em si mesmos", isto é, em considerar suas máquinas como sistemas totalmente operacionais, completos, como hardware e software. Em geral, os usuários de um SORD não dispõem da opção de escolher outros produtos de software para sua máquina.

A companhia buscou corrigir essa situação fornecendo o sistema operacional SB-80 como opção para sua série M23 de computadores baseados no Z80. O SB-80 equivale ao sistema operacional CP/M 2.2, da Digital Research, e permite o uso do software relacionado ao CP/M nos computadores SORD. Outro passo da empresa nesse sentido foi tornar seus equipamentos compatíveis com o p-System, elaborado na

Universidade da Califórnia, em San Diego (UCSD).

Para os usuários satisfeitos em limitar-se ao software da própria SORD, a companhia oferece vários BASICs potentes, seu próprio sistema operacional, o software integrado PIPS e um processador de textos.

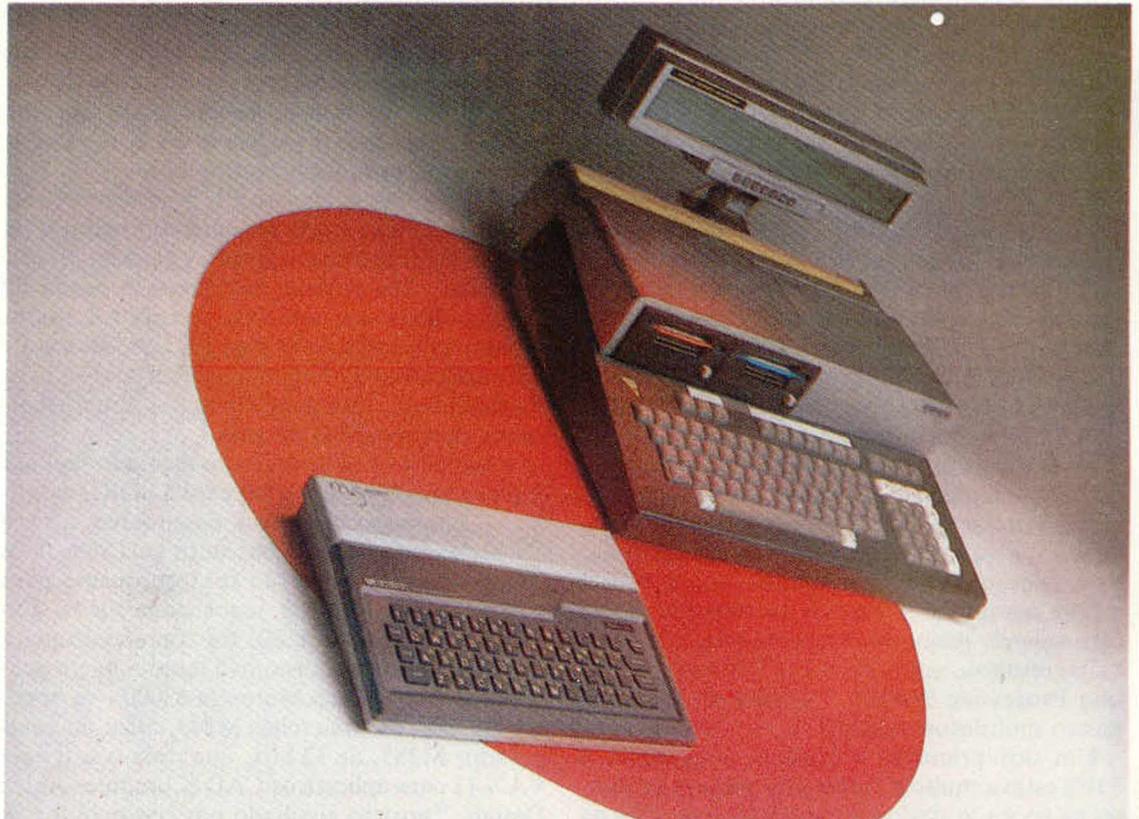
Além de dar acesso ao CP/M e ao p-System, a SORD tende a seguir a trilha aberta por fabricantes de minicomputadores, que oferecem sistemas dependentes de seu próprio software. Enquanto isso, as indústrias de computação restantes descobriram que o caminho para melhorar as vendas consiste na padronização e na compatibilidade entre famílias de computadores. A SORD tem revelado lentidão em se adaptar a essa tendência. Continua a desenvolver projetos inovadores de hardware, mas, dizem os críticos, falha em não apoiar os usuários com documentação adequada para seus produtos.

Desde 1983 a SORD tem se esforçado em elaborar boa documentação para o PIPS e seu processador de textos. Mas, mesmo em 1985, faltava documentação mais aprofundada de seu hardware, e notava-se sua relutância em encorajar outros fabricantes de software a desenvolver produtos que rodassem em seus computadores.

Os próximos anos serão críticos para a SORD, que se defrontará com a IBM e com o poderio industrial dos maiores fabricantes de computadores do Japão. Seu presidente é um fervoroso defensor da iniciativa individual e da pequena empresa, e teve sucesso em fazer da SORD uma companhia internacional. Será ela capaz de manter-se no mundo da informática?

Opções SORD

As máquinas da SORD mais conhecidas são o computador pessoal M5 e o portátil comercial M23P. Este último impôs novos padrões para os portáteis ao usar discos flexíveis Sony e um visor de cristal líquido de 80 colunas.



```

910 REM *****MONSTROS GALACTICOS*****
920 LET B=INT(RND(1)*10)
930 LET B=INT(B*10)
940 LET B=INT(B*10)
950 LET B=INT(B*10)
960 LET B=INT(B*10)
970 LET B=INT(B*10)
980 LET B=INT(B*10)
990 LET B=INT(B*10)

```

MONSTROS GALACTICOS PARA LINHA APPLE

Você precisa atravessar os quadrantes do espaço com sua nave. Mas tome cuidado para não colidir com os seres alienígenas que habitam algumas regiões. E não adianta tentar atingi-los, pois são indestrutíveis. Será bem melhor tentar a travessia desviando-se deles. Não pense que será fácil: a cada tela percorrida, o número de inimigos aumenta e eles estarão cada vez mais inquietos. Difícilmente você chegará ao outro lado incólume. Uma ou outra vez, acabará esbarrando num deles. Mas lembre-se: sua nave só agüenta cinco colisões.

```

1000 REM*****
1010 REM**
1020 REM**
1030 REM**
1040 REM**
1050 GOSUB 2110
1060 DIM M(8,8),MX(8),MY(8)
1070 BE$=CHR$(7)
1080 HOME
1090 GOTO 1700
1100 REM ** PLOTA TABELA DE FORMAS **
1110 HCOLOR=3:DRAW SH AT 46+19*X,1+19*Y
1120 RETURN
1130 REM ** LIMPA TABELA DE FORMAS **
1140 HCOLOR=0:DRAW SH AT 46+19*X,1+19*Y
1150 RETURN
1160 REM ** MOVIMENTO DO JOGADOR **
1170 PRINT BE$:
1180 MV=PEEK(-16384)-128
1190 IF MV>0 THEN FOK$="R" AND PX<8:-(MV=ASC("J")) AND PX>1
1210 MY=(MV=ASC("M")) AND PY<8:-(MV=ASC("I")) AND PY>1
1220 IF MX<0 OR MY<0 THEN SH=1:X=PX:Y=PY:GOSUB 1130
1230 IF M(PX,PY)>0 THEN SH=2:GOSUB 1100
1240 X=PX+MX:Y=PY+MY:SH=1:GOSUB 1100
1250 PX=X:PY=Y
1260 RETURN
1270 REM ** JOGADOR E APANHADO **
1280 FOR TI=1 TO M(PX,PY)
1290 PRINT BE$+BE$+BE$+BE$+BE$:
1300 NEXT
1310 NF=NF+M(PX,PY)
1320 RETURN
1330 REM ** MOVIMENTO DO MONSTRO **
1340 SH=2
1350 FOR MO=1 TO 8
1360 X=MX(MO):Y=MY(MO)
1370 IF INT(RND(1)*10)<NC THEN 1430
1380 REM ** MOVIMENTO ALEATORIO **
1390 DJ=INT(RND(1)*4)
1400 MX=(DI=0 AND X<8)-(DI=1 AND X>1)
1410 MY=(DI=2 AND Y<8)-(DI=3 AND Y>1)

```

```

1840 M(1,8)=1
1870 NEXT
1880 PRINT BE$+BE$+BE$:
1890 RETURN
1900 REM ** PROGRAMA PRINCIPAL **
1910 GOSUB 1610
1920 GOSUB 1690
1930 GOSUB 1160
1940 SC=SC-1
1950 VTAB 22:HTAB 16:PRINT "PONTOS: "SC:
1960 IF M(PX,PY)=0 THEN 2000
1970 GOSUB 1270
1980 IF NF>4 THEN 2080
1990 GOTO 1920
2000 IF PY<8 THEN 2050
2010 NC=NC+1
2020 SC=SC+20*NC-5*NF
2030 VTAB 22:HTAB 16:PRINT "PONTOS: "SC:
2040 GOTO 1920
2050 GOSUB 1330
2060 GOTO 1930
2070 END
2080 TEXT :HOME:VTAB 14:HTAB 20:PRINT "
2090 FIN"
2100 REM ** INSTRUÇÕES **
2110 HOME:VTAB 2:HTAB 10:PRINT "MONSTRO
S GALACTICOS"
2120 PRINT "SUA MISSAO E' A DE

```

GUERRA NAS ESTRELAS PARA TK85

Você está no comando de uma nave espacial Sarggon X que se encontra na parte baixa da atmosfera de Tiffalian, planeta de um sistema distante cerca de 240 anos-luz da Terra. Nele há uma base WGK, uma base de combate da guerra nas estrelas. Seu objetivo é conduzir a Sarggon X para essa base, famosa pelo poder de destruição. No caminho, você encontrará naves Uranium, inofensivas desde que não sejam tocadas. Devido à sua forma, a Sarggon X possui controles bem simples, que são as teclas das setas ou o joystick. Boa sorte!

```

910 REM *****PROGRAMA PARA O TK 85*****
920 LET X=10
930 LET Y=10
940 LET H=PEEK 16396+PEEK 16397
950 IF H=PEEK 16396+PEEK 16397
960 LET F=PEEK (A+Y*33+X+33)<>0 OR
PEEK (A+Y*33+X+34)<>0 OR PEEK (
A+Y*33+X+35)<>0 THEN GOTO 50
970 NEXT N
980 GOTO 120
990 PRINT AT Y,X:
1000 PRINT AT Y,X:
1010 PRINT AT Y,X:
1020 PRINT AT Y,X:
1030 PRINT AT Y,X:
1040 PRINT AT Y,X:
1050 PRINT AT Y,X:
1060 PRINT AT Y,X:
1070 PRINT AT Y,X:
1080 PRINT AT Y,X:
1090 PRINT AT Y,X:
1100 PRINT AT Y,X:
1110 PRINT AT Y,X:
1120 PRINT AT Y,X:
1130 PRINT AT Y,X:
1140 PRINT AT Y,X:
1150 PRINT AT Y,X:
1160 PRINT AT Y,X:
1170 PRINT AT Y,X:
1180 PRINT AT Y,X:
1190 PRINT AT Y,X:
1200 PRINT AT Y,X:
1210 PRINT AT Y,X:
1220 PRINT AT Y,X:
1230 PRINT AT Y,X:
1240 PRINT AT Y,X:
1250 PRINT AT Y,X:
1260 PRINT AT Y,X:
1270 PRINT AT Y,X:
1280 PRINT AT Y,X:
1290 PRINT AT Y,X:
1300 PRINT AT Y,X:
1310 PRINT AT Y,X:
1320 PRINT AT Y,X:
1330 PRINT AT Y,X:
1340 PRINT AT Y,X:
1350 PRINT AT Y,X:
1360 PRINT AT Y,X:
1370 PRINT AT Y,X:
1380 PRINT AT Y,X:
1390 PRINT AT Y,X:
1400 PRINT AT Y,X:
1410 PRINT AT Y,X:
1420 PRINT AT Y,X:
1430 PRINT AT Y,X:
1440 PRINT AT Y,X:
1450 PRINT AT Y,X:
1460 PRINT AT Y,X:
1470 PRINT AT Y,X:
1480 PRINT AT Y,X:
1490 PRINT AT Y,X:
1500 PRINT AT Y,X:
1510 PRINT AT Y,X:
1520 PRINT AT Y,X:
1530 PRINT AT Y,X:
1540 PRINT AT Y,X:
1550 PRINT AT Y,X:
1560 PRINT AT Y,X:
1570 PRINT AT Y,X:
1580 PRINT AT Y,X:
1590 PRINT AT Y,X:
1600 PRINT AT Y,X:
1610 PRINT AT Y,X:
1620 PRINT AT Y,X:
1630 PRINT AT Y,X:
1640 PRINT AT Y,X:
1650 PRINT AT Y,X:
1660 PRINT AT Y,X:
1670 PRINT AT Y,X:
1680 PRINT AT Y,X:
1690 PRINT AT Y,X:
1700 PRINT AT Y,X:
1710 PRINT AT Y,X:
1720 PRINT AT Y,X:
1730 PRINT AT Y,X:
1740 PRINT AT Y,X:
1750 PRINT AT Y,X:
1760 PRINT AT Y,X:
1770 PRINT AT Y,X:
1780 PRINT AT Y,X:
1790 PRINT AT Y,X:
1800 PRINT AT Y,X:
1810 PRINT AT Y,X:
1820 PRINT AT Y,X:
1830 PRINT AT Y,X:
1840 PRINT AT Y,X:
1850 PRINT AT Y,X:
1860 PRINT AT Y,X:
1870 PRINT AT Y,X:
1880 PRINT AT Y,X:
1890 PRINT AT Y,X:
1900 PRINT AT Y,X:
1910 PRINT AT Y,X:
1920 PRINT AT Y,X:
1930 PRINT AT Y,X:
1940 PRINT AT Y,X:
1950 PRINT AT Y,X:
1960 PRINT AT Y,X:
1970 PRINT AT Y,X:
1980 PRINT AT Y,X:
1990 PRINT AT Y,X:
2000 PRINT AT Y,X:
2010 PRINT AT Y,X:
2020 PRINT AT Y,X:
2030 PRINT AT Y,X:
2040 PRINT AT Y,X:
2050 PRINT AT Y,X:
2060 PRINT AT Y,X:
2070 PRINT AT Y,X:
2080 PRINT AT Y,X:
2090 PRINT AT Y,X:
2100 PRINT AT Y,X:
2110 PRINT AT Y,X:
2120 PRINT AT Y,X:
2130 PRINT AT Y,X:
2140 PRINT AT Y,X:
2150 PRINT AT Y,X:
2160 PRINT AT Y,X:
2170 PRINT AT Y,X:
2180 PRINT AT Y,X:
2190 PRINT AT Y,X:
2200 PRINT AT Y,X:
2210 PRINT AT Y,X:
2220 PRINT AT Y,X:
2230 PRINT AT Y,X:
2240 PRINT AT Y,X:
2250 PRINT AT Y,X:
2260 PRINT AT Y,X:
2270 PRINT AT Y,X:
2280 PRINT AT Y,X:
2290 PRINT AT Y,X:
2300 PRINT AT Y,X:
2310 PRINT AT Y,X:
2320 PRINT AT Y,X:
2330 PRINT AT Y,X:
2340 PRINT AT Y,X:
2350 PRINT AT Y,X:
2360 PRINT AT Y,X:
2370 PRINT AT Y,X:
2380 PRINT AT Y,X:
2390 PRINT AT Y,X:
2400 PRINT AT Y,X:
2410 PRINT AT Y,X:
2420 PRINT AT Y,X:
2430 PRINT AT Y,X:
2440 PRINT AT Y,X:
2450 PRINT AT Y,X:
2460 PRINT AT Y,X:
2470 PRINT AT Y,X:
2480 PRINT AT Y,X:
2490 PRINT AT Y,X:
2500 PRINT AT Y,X:
2510 PRINT AT Y,X:
2520 PRINT AT Y,X:
2530 PRINT AT Y,X:
2540 PRINT AT Y,X:
2550 PRINT AT Y,X:
2560 PRINT AT Y,X:
2570 PRINT AT Y,X:
2580 PRINT AT Y,X:
2590 PRINT AT Y,X:
2600 PRINT AT Y,X:
2610 PRINT AT Y,X:
2620 PRINT AT Y,X:
2630 PRINT AT Y,X:
2640 PRINT AT Y,X:
2650 PRINT AT Y,X:
2660 PRINT AT Y,X:
2670 PRINT AT Y,X:
2680 PRINT AT Y,X:
2690 PRINT AT Y,X:
2700 PRINT AT Y,X:
2710 PRINT AT Y,X:
2720 PRINT AT Y,X:
2730 PRINT AT Y,X:
2740 PRINT AT Y,X:
2750 PRINT AT Y,X:
2760 PRINT AT Y,X:
2770 PRINT AT Y,X:
2780 PRINT AT Y,X:
2790 PRINT AT Y,X:
2800 PRINT AT Y,X:
2810 PRINT AT Y,X:
2820 PRINT AT Y,X:
2830 PRINT AT Y,X:
2840 PRINT AT Y,X:
2850 PRINT AT Y,X:
2860 PRINT AT Y,X:
2870 PRINT AT Y,X:
2880 PRINT AT Y,X:
2890 PRINT AT Y,X:
2900 PRINT AT Y,X:
2910 PRINT AT Y,X:
2920 PRINT AT Y,X:
2930 PRINT AT Y,X:
2940 PRINT AT Y,X:
2950 PRINT AT Y,X:
2960 PRINT AT Y,X:
2970 PRINT AT Y,X:
2980 PRINT AT Y,X:
2990 PRINT AT Y,X:
3000 PRINT AT Y,X:

```

